ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Mechanistic Interpretability in Text-to-Image Diffusion Models

Master Thesis

Jan Heinrich Schlegel

Department of Mathematics,
Seminar for Statistics, ETH Zürich

September 8, 2025

Supervisor: Prof. Dr. Julia E. Vogt
Advisors: Moritz Vandenhirtz, Alice Bizeul

Department of Computer Science, Institute for Machine Learning, ETH Zürich

Abstract

Despite the widespread success of text-to-image diffusion models, their internal mechanisms remain largely opaque. In this work, we mechanistically analyze Stable Diffusion v2.1 by applying Sparse Autoencoders (SAEs) to decompose the dense activations of its cross-attention blocks into a sparse, interpretable feature basis. Through time-stratified causal interventions, we reveal a clear temporal hierarchy in the generative process, demonstrating that early timesteps are critical for defining an image's global composition and style, the mid-stage refines the established structure and blends concepts, while later stages perform textural refinement. Moreover, we find that the learned feature dictionary exhibits remarkable cross-resolution generalization, maintaining intervention and reconstruction efficacy across different image sizes, and suggesting that SAEs can be trained more efficiently on lower-resolution data without sacrificing their analytical power. Moving beyond single-feature analysis, we introduce a novel framework for discovering temporal computational subgraphs in diffusion models. By adapting gradient-based attribution techniques, we trace causal dependencies between features across early denoising steps. By identifying these causal computational pathways that link sequences of these interpretable features, our work provides a deeper mechanistic understanding of the generative algorithm, paving the way for more controllable and reliable generative models.

Contents

Chapter 1

Introduction

Diffusion models have established themselves at the forefront of text-to-image generation [52]. Despite their widespread success, the internal mechanisms by which they gradually translate textual inputs into high-resolution visual outputs remain largely opaque. This lack of interpretability poses a significant barrier to the safe and reliable adoption of diffusion models in high-stakes environments, such as medical imaging, where precise control over and understanding of the model's outputs are paramount. The high dimensionality, multimodal nature, and iterative denoising process of these models make interpretability in text-to-image diffusion particularly challenging [20, 26].

Mechanistic interpretability has emerged as a promising direction to tackle this obstacle. Unlike traditional explainability techniques that primarily examine input-output relationships, mechanistic interpretability seeks to decode complex neural networks by reverse-engineering their intermediate representations into human-understandable concepts [3]. A central challenge in understanding neural networks intrinsically is that single neurons frequently encode multiple unrelated concepts, a phenomenon referred to as polysemanticity [11, 40]. Sparse autoencoders (SAEs), a scalable approach to sparse dictionary learning, directly address this problem. SAEs aim to discover a sparse overcomplete basis consisting of monosemantic features, where each feature ideally corresponds to a single, human-interpretable concept. This SAE approach has demonstrated success in both large language models [4, 47, 5] and vision models [14].

In the realm of text-to-image diffusion models, recent studies have shown that SAEs can extract interpretable and causally influential features from key architectural components, including the diffusion model's cross-attention blocks [6, 35, 23, 42, 49], and the text encoder [22]. These efforts have enabled powerful downstream applications, such as concept unlearning [6], steering generations towards safer modes [22], and concept transfer [46].

Furthermore, methods are being developed to make SAEs more efficient and temporally aware, for instance by training a single SAE that operates across all timesteps [6, 46, 42] or by additionally introducing learnable, timestep-specific modulation [20].

While these advances are promising, they primarily leverage SAE features as tools for downstream applications. Consequently, a comprehensive mechanistic understanding of how these features interact throughout the multi-step generation process remains underexplored. This thesis directly addresses this gap by performing a mechanistic analysis of two cross-attention blocks in a v2.1 checkpoint of the Stable Diffusion model [37]. Our work moves beyond the analysis of single SAE features to investigate the computational circuits that govern their interactions during image synthesis. Ultimately, understanding these circuits is the key to explaining how a diffusion model progressively transforms random noise into a coherent final image, paving the way for more controllable and reliable generative models.[1]

The primary contributions of this thesis are as follows:

1. We provide a systematic temporal analysis of sparse autoencoder feature roles in Stable Diffusion (SD) v2.1 [37], showing through time-stratified interventions that in the early-stage features control global composition and style, the mid-stage refines the established structure and blends concepts, while in the late-stage features primarily refine texture.
2. We demonstrate that SAE features learned at 512×512 resolution generalize effectively to 768×768, maintaining both reconstruction quality and intervention efficacy across resolutions.
3. We introduce a novel temporal circuit discovery framework for diffusion models, using gradient-based attribution and a diagnostic probe-based objective to trace causal feature dependencies across timesteps.
4. We qualitatively validate discovered circuits using a novel bidirectional activation patching method. Our approach represents a key technical extension of Surkov et al.'s [46] activation patching implementation to the challenging setting of non-distilled diffusion models that employ classifier-free guidance. We demonstrate successful concept transformation when swapping identified feature sets on a curated, simplistic dataset, providing initial evidence for the task-relevance of the discovered circuit mechanisms.

This work is structured to systematically build this mechanistic understanding of SD v2.1. Chapter 2 reviews the relevant literature on interpretability

[1] The code for this project is available at https://github.com/JHSchlegel/mechinterp-diffusion

in diffusion models and provides a primer on circuit discovery in LLMs. Chapter 3 then establishes the necessary theoretical framework, detailing the mathematical notation and the core architectures used throughout this work. In Chapter 4, we present our trained SAE and demonstrate how we can utilize its feature dictionary to causally steer image generation at different stages of the diffusion process. Building on this, Chapter 5 presents our core contribution in the form of an analysis of the interplay between SAE features to form primitive computational circuits. Chapter 6 summarizes the key findings of the thesis, discusses their broader implications for the fields of interpretability, AI safety, and generative modeling, and outlines promising directions for future research. Finally, in Chapter 7 we conclude.

Chapter 2

Related Work

To contextualize our contributions to mechanistic interpretability in diffusion models, we review the fast-growing body of work that applies sparse autoencoders to understand and control text-to-image generation. We first examine how recent studies successfully extract interpretable features from various components of diffusion model architectures. Subsequently, we detail how these features are leveraged for downstream applications such as concept unlearning and bias mitigation. Finally, we discuss emerging efforts in circuit discovery, a field primarily developed for large language models but which offers promising methodologies that we adapt to analyze feature interactions within diffusion models.

Surkov et al. [46] pioneered the application of SAEs to text-to-image models, applying them to the cross-attention blocks of the distilled, few-step diffusion models SDXL-Turbo [39] and Flux-Schnell [28]. They reveal specializations across different architectural components and demonstrate the causal influence of learned SAE features on the final image through targeted interventions. Additionally, they systematize these interventions within their RIEBench [46] benchmark, which facilitates spatially-constrained, object-level feature transfer requiring segmentation masks. In particular, their method implements a form of conceptual steering, where they identify the most discriminative feature indices for a concept and activate them within a segmentation mask using a new, statistically-derived magnitude instead of the original activation values. Moreover, they find that these learned SAE features are surprisingly portable between models. Specifically, features trained exclusively on single-step SDXL-Turbo effectively transfer to 4-step SDXL-Turbo and to the base multi-step SDXL model [35], enabling causal edits without requiring retraining.

Building on this foundation, Kim et al. [23] employ separate SAEs for different timesteps and layers to investigate how visual information is encoded at various diffusion stages and blocks. Their primary finding is that the

granularity of visual features in diffusion models varies non-linearly across the model's depth and the model architecture. Additionally, they emphasize that the learned SAE features exhibit strong predictive performance in combination with lightweight classifiers on downstream classification tasks. Delving deeper into the temporal dynamics, Tinaz et al. [49] apply SAEs to investigate the internal representations of Stable Diffusion v1.4 [37] at different stages of the generation process. Specifically, they train separate SAEs on activations from early ($t = 1.0$), middle ($t = 0.5$), and late ($t = 0.0$) snapshots of the generative process. Through causal interventions on the discovered SAE concepts, they establish that image composition can be effectively controlled in the early stages and image style in the middle stages. Ultimately, the final stages are reserved for minor textural refinement.

While the aforementioned studies probe the internal workings of a diffusion U-Net [38] or transformer [50], subsequent work by Kim and Ghadiyaram [22] demonstrates that SAEs trained on the activations of the text encoder of Stable Diffusion v1.4 [37] can effectively steer diffusion generation toward safer outputs. For example, their SAE is capable of removing nudity or introducing a new concept, such as an image style, without compromising overall image fidelity. Notably, this method modifies only the text embeddings rather than intervening directly in the U-Net's cross-attention blocks.

Complementing these text-level interventions, other research achieves similar safety and alignment goals by operating directly on the cross-attention blocks within the U-Net. For instance, Cywiński and Deja [6] apply this principle to concept unlearning in Stable Diffusion v1.5 [37]. Using a single SAE trained across all timesteps, they employ a custom score function to identify features highly specific to a target concept. They then ablate these high-scoring features during inference. This feature surgery effectively removes an unwanted object or style, achieves state-of-the-art results on the UnlearnCanvas benchmark [54], and demonstrates high robustness. Extending this principle to bias mitigation, Shi et al. [42] present a framework that utilizes a gradient-based method to pinpoint SAE features causally linked to social biases, such as those related to gender, age, and race. By directly intervening on these identified "bias features," the method allows for precise, fine-grained control over the bias level in the generated content while leaving other semantic attributes largely untouched.

The finding that complex tasks like concept unlearning [6] and transfer [46] often require the simultaneous manipulation of a constellation of multiple SAE features points toward a limitation of single-feature analysis. This necessity to consider interactions between features motivates the field of circuit discovery, which aims to identify the specific computational subgraphs within a neural network responsible for a particular behaviour. Circuit discovery in large language models originated with neuron-level analysis in

foundational work by Elhage et al. [12], who systematically mapped attention patterns and MLP neurons in small transformers to uncover mechanisms, such as induction heads, that drive in-context learning. Wang et al. [51] advance these methods through path patching, which causally isolates computational pathways linking model components to specific outputs.

While these approaches demonstrate that model behaviors arise from circuits, the polysemantic nature of individual neurons motivated a shift toward feature-level analysis. Marks et al. [31], for instance, introduce SHIFT, a scalable framework that automates circuit discovery by operating directly on SAE features rather than neurons. Their method employs gradient-based attribution to identify causally essential features and then traces the computational graph to uncover the minimal circuit responsible for a given task. Similarly, Ameisen et al. [1] employ sparse transcoders [29, 10] that learn transformations between consecutive layers while preserving the complete computational graph, enabling direct circuit tracing through an interpretable feature space. Uncovering analogous circuits in diffusion models, for instance, understanding how features for ”cat”, ”fur”, and ”sitting” combine across timesteps to form a coherent image, remains a critical open challenge that this thesis begins to address.

Chapter 3

Background

This chapter establishes the theoretical foundation for the analysis presented in this thesis. We begin with a comprehensive overview of diffusion models, covering their core principles, their evolution into latent diffusion models, and the critical role of cross-attention in conditional image generation. We then introduce the analytical framework used to isolate and study the updates within these cross-attention blocks. Subsequently, we address the core challenge of interpreting these dense neural representations by discussing the superposition hypothesis and introducing sparse dictionary learning as a powerful method for disentangling them. Finally, we detail the specific sparse autoencoder architecture and training objective that we employ in our experiments to render the model's computations interpretable.

3.1 Diffusion Models

Diffusion models are a class of generative models that learn to reverse a gradual stochastic diffusion process. This framework consists of a predefined forward process that incrementally perturbs the data, transforming it into a simple prior distribution, and a learned reverse process designed to invert this transformation, generating new samples from the data distribution.

The forward diffusion process $q(\cdot)$ is defined as a Markov chain of T timesteps that progressively diffuses a data sample $\boldsymbol{x_0} \sim q(\boldsymbol{x_0})$ into isotropic Gaussian noise. Each transition follows

$$q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{1-\beta_t}\boldsymbol{x}_{t-1}, \beta_t \boldsymbol{I}),$$

where $\{\beta_t\}_{t=1}^T$ denotes the noise variance schedule with $\beta_t \in (0,1)\ \forall\ t \in [T]$ with $[T] = \{1, \ldots, T\}$. Ho et al. [17] show that, due to the Markovian struc-

ture, we can sample $\boldsymbol{x}_t$ directly from $\boldsymbol{x}_0$ without intermediate computations. Defining $\bar{\alpha}_t = \prod_{\tau=1}^{t}(1-\beta_\tau)$, we thus obtain

$$q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, (1-\bar{\alpha}_t)\boldsymbol{I}),$$

and its reparameterization

$$\boldsymbol{x}_t = \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(0, \boldsymbol{I}).$$

The reverse process learns to invert this forward diffusion. It hence generates samples starting from pure noise, by approximating the intractable posterior distribution $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ with a parametrized Gaussian distribution $p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$. Instead of directly parameterizing the mean and variance of this conditional distribution, modern diffusion models train a neural network $\boldsymbol{\varepsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$ to predict the noise component $\boldsymbol{\varepsilon}$ from the noisy input $\boldsymbol{x}_t$ at a given timestep t [17]. This parametrization leads to a simplified training objective derived from maximizing the variational lower bound (ELBO) on the log-likelihood:

$$\mathcal{L}(\boldsymbol{\theta}) = \mathbb{E}_{t,\boldsymbol{x}_0,\boldsymbol{\varepsilon}}\left[\|\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_{\boldsymbol{\theta}}(\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\varepsilon}, t)\|^2\right].$$

This formulation is closely connected to score matching [44], where the model learns to estimate the score, defined as the gradient of the log-probability density of the perturbed data distribution. For this reason, modern implementations adopt a continuous-time formulation, which provides a unified framework for various diffusion models and enables flexible sampling schemes with arbitrary numbers of steps. In this formulation, $t = 0.0$ represents uncorrupted data and $t = 1.0$ pure noise. We follow this time convention throughout.

To guide this generative process, conditional generation incorporates additional information $\boldsymbol{c}$ by modifying the reverse process to $p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{c})$. This conditioning signal is typically integrated via cross-attention mechanisms [50]:

$$\text{CrossAttention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{d_k}}\right)\boldsymbol{V},$$

where queries $\boldsymbol{Q}$ derive from image features, keys $\boldsymbol{K}$ and values $\boldsymbol{V}$ are projections from the conditioning embeddings, and d_k is the dimensionality of the key vectors.

A fundamental limitation of diffusion models lies in their computational demands when operating directly in high-dimensional pixel space. The prohibitive cost of training and inference on raw pixel values necessitates more

efficient approaches. Latent diffusion models [37] address this challenge by performing the diffusion process in a lower-dimensional latent space learned through a Variational Autoencoder (VAE) [25]. Specifically, an encoder $\mathcal{E}$ compresses an image $\boldsymbol{x}$ into a latent representation $\boldsymbol{z} = \mathcal{E}(\boldsymbol{x})$, while a decoder reconstructs the image from the denoised latent code.

Stable Diffusion (SD) v2.1 exemplifies this latent diffusion approach, utilizing a U-Net [38] to parametrize the noise prediction network $\boldsymbol{\varepsilon_\theta}$. As illustrated in Figure 3.1, this U-Net incorporates an asymmetrical arrangement of cross-attention blocks for text conditioning via CLIP embeddings [36]. These cross-attention blocks, which inject conditioning information at multiple scales throughout the U-Net, are the focus of our interpretability analysis.

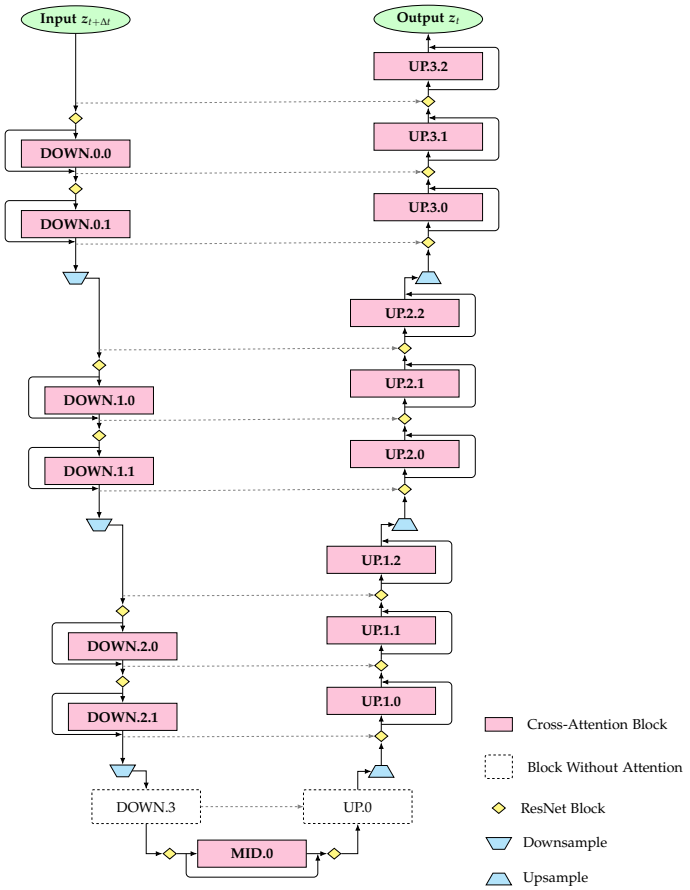


Figure 3.1: Stable Diffusion v2.1 U-Net Architecture. The encoder (left) contains two ResNet [15] and cross-attention modules per resolution level, except for the deepest level, which omits attention. In contrast, the decoder (right) is asymmetric, using three such modules per block level. This architectural design, which maps a noisy latent representation $z_{t+\Delta t}$ to z_t, allows for robust injection of conditioning information across multiple scales. Own illustration inspired by Figure 10 in Surkov et al. [46].

3.2 Cross-Attention Analysis Framework

To formalize our analysis, we adopt the notation of Surkov et al. [46] and define $\mathcal{T}_{\ell,t} : \mathbb{R}^{h\times w\times d} \times \mathcal{C} \to \mathbb{R}^{h\times w\times d}$ as the transformation performed by the ℓ^{th} cross-attention block at timestep t, where $\mathcal{C}$ denotes the conditioning space containing text embeddings. The residual update is

$$\boldsymbol{\Delta D}_{\ell,t}(\boldsymbol{c}, \boldsymbol{z}_t) := \mathcal{T}_{\ell,t}(\boldsymbol{D}^{\text{in}}_{\ell,t}(\boldsymbol{c}, \boldsymbol{z}_t), \boldsymbol{c}) = \boldsymbol{D}^{\text{out}}_{\ell,t}(\boldsymbol{c}, \boldsymbol{z}_t) - \boldsymbol{D}^{\text{in}}_{\ell,t}(\boldsymbol{c}, \boldsymbol{z}_t),$$

where $\boldsymbol{D}^{\text{in}}(\boldsymbol{c}, \boldsymbol{z}_t), \boldsymbol{D}^{\text{out}}(\boldsymbol{c}, \boldsymbol{z}_t) \in \mathbb{R}^{h\times w\times d}$ represent the latent representations before and after application of the cross-attention transformer block respectively, $\boldsymbol{c} \in \mathcal{C}$ is the text embedding, and $\boldsymbol{z}_t$ is the input noise. For notational convenience, moving forward, we will neglect the dependence on $\boldsymbol{c}$, $\boldsymbol{z}_t$, and ℓ when the context is unambiguous. At every spatial coordinate (i, j), we can write the residual update vector $\boldsymbol{\Delta D}_{ij,t} \in \mathbb{R}^d$ as

$$\boldsymbol{\Delta D}_{ij,t} := \mathcal{T}(\boldsymbol{D}^{\text{in}}_t, \boldsymbol{c})_{ij} = \boldsymbol{D}^{\text{out}}_{ij,t} - \boldsymbol{D}^{\text{in}}_{ij,t}.$$

These residual updates, $\boldsymbol{\Delta D}_{ij,t}$, precisely capture how conditioning information affects the generation at different spatial locations and timesteps. However, these vectors are dense and high-dimensional, posing a significant challenge for direct interpretation. The following sections describe our methodology for decomposing these dense vectors into a sparse, interpretable basis.

3.3 Sparse Dictionary Learning

The primary obstacle to interpreting dense representations like $\boldsymbol{\Delta D}_{ij,t}$ is the phenomenon of polysemanticity: when a single neuron activates in response to numerous, unrelated concepts. The leading explanation for polysemanticity is the superposition hypothesis [11], which proposes that networks compress a vast set of sparse features into a lower-dimensional space by embedding each such feature as a nearly orthogonal direction. This hypothesis is justified by principles such as the Johnson-Lindenstrauss Lemma (cf. Theorem A.1), which allows for a potentially exponential number of almost-orthogonal vectors to coexist. In other words, a network exploiting these almost-orthogonal directions can store far more concepts than neurons, inevitably creating polysemantic units.

The most successful approach to resolve this entanglement is sparse dictionary learning in the form of Sparse Autoencoders (SAEs). An SAE aims to reverse the process of superposition by decomposing polysemantic activation vectors into sparse combinations of monosemantic features from a learned, overcomplete dictionary. In our case, we train an SAE to transform each dense residual update vector $\boldsymbol{\Delta D}_{ij,t} \in \mathbb{R}^d$ from the U-Net's cross-attention blocks into a high-dimensional but sparse representation $\boldsymbol{S}_{ij,t} \in \mathbb{R}^{n_f}$, where

typically $n_f >> d$ to provide sufficient capacity to assign distinct concepts to distinct feature dimensions.

Concretely, an SAE is an unsupervised model with a sparse bottleneck. It consists of an encoder that maps a dense input vector $\boldsymbol{\Delta D}_{ij,t}$ to a sparse representation $\boldsymbol{S}_{ij,t}$, and a decoder that reconstructs the original vector from this sparse code. The decoder employs learned feature vectors $\boldsymbol{f}_\varphi \in \mathbb{R}^d$ to perform the reconstruction

$$\boldsymbol{\Delta D}_{ij,t} \approx \widehat{\boldsymbol{\Delta D}_{ij,t}} = \sum_{\varphi=1}^{n_f} \boldsymbol{S}_{ij,t}^{\varphi} \boldsymbol{f}_\varphi,$$

where $\boldsymbol{S}_{ij,t}^{\varphi} \in \mathbb{R}$ is the scalar activation of the φ-th feature. Both encoder and decoder parameters are shared across all spatial positions and timesteps.

3.4 TopK Sparse Autoencoders

Building upon the general framework of sparse dictionary learning, the specific SAE architecture we employ is a k-sparse autoencoder [30], illustrated in Figure 3.2. While we previously described operations on individual residual updates $\boldsymbol{\Delta D}_{ij,t}$, in practice, we train on mini-batches of these vectors, which are collected across different spatial positions, timesteps, and prompts. For notational simplicity, we denote a generic input vector to the SAE as $\boldsymbol{v} \in \mathbb{R}^d$ and its sparse code (denoted as $\boldsymbol{S}_{ij,t}$ for a single location and timestep) as $\boldsymbol{h} \in \mathbb{R}^{n_f}$. We obtain this sparse hidden representation through an encoder, that is,

$$\boldsymbol{h} = \mathrm{ENC}(\boldsymbol{v}) = \text{top-k}(\mathrm{ReLU}(\boldsymbol{W}_{\text{enc}}(\boldsymbol{v} - \boldsymbol{b}_{\text{pre}}) + \boldsymbol{b}_{\text{enc}})),$$

where $\boldsymbol{W}_{\text{enc}} \in \mathbb{R}^{n_f \times d}$ is the encoder weight matrix, $\boldsymbol{b}_{\text{pre}} \in \mathbb{R}^d$ is the pre-encoder bias, and $\boldsymbol{b}_{\text{enc}} \in \mathbb{R}^{n_f}$ the encoder bias. The top-k$(\cdot)$ activation function enforces sparsity by zeroing all but the k largest input values, thereby ensuring that $\|\boldsymbol{h}\|_0 \leq k$.

The decoder then reconstructs the input from this sparse code $\boldsymbol{h}$ through

$$\widehat{\boldsymbol{v}} = \mathrm{DEC}(\boldsymbol{v}) = \boldsymbol{W}_{\text{dec}} \boldsymbol{h} + \boldsymbol{b}_{\text{pre}},$$

where we initialize the decoder weights $\boldsymbol{W}_{\text{dec}} = [\boldsymbol{f}_1, \boldsymbol{f}_2, \ldots, \boldsymbol{f}_{n_f}] \in \mathbb{R}^{d \times n_f}$ as the transpose of $\boldsymbol{W}_{\text{enc}}$, where each column $\boldsymbol{f}_\varphi$ represents a learned feature vector.

A widespread practical challenge in training large SAEs is the formation of dead features [13, 47], i.e., decoder columns that remain inactive across large portions of training data. Dead features represent unused capacity and

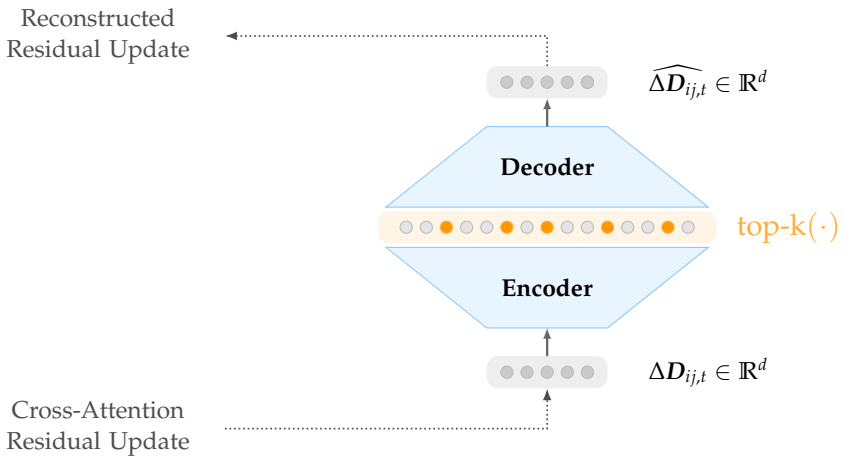


Figure 3.2: The TopK Sparse Autoencoder Architecture. The SAE learns a sparse, overcomplete basis for activation vectors $\Delta D_{ij,t} \in \mathbb{R}^d$ from a cross-attention block. An encoder projects $\Delta D_{ij,t}$ into a higher-dimensional latent space. Sparsity is enforced via a top-k$(\cdot)$ operator, which retains only the k features with the largest activations. Finally, a decoder linearly projects these active features back to the original dimension to produce the reconstruction $\widehat{\Delta D_{ij,t}} \in \mathbb{R}^d$.

effectively reduce the dimensionality of the learned dictionary. To mitigate this issue, we augment the primary reconstruction loss with an auxiliary loss that encourages dead features to activate and represent unexplained variance. Specifically, we model the reconstruction error $\boldsymbol{e} = \boldsymbol{v} - \widehat{\boldsymbol{v}}$ using an auxiliary set of features:

$$\begin{aligned} \boldsymbol{h}_{\text{aux}} &= \text{top-k}_{\text{aux}}(\text{ReLU}(\boldsymbol{W}_{\text{enc}}(\boldsymbol{e} - \boldsymbol{b}_{\text{pre}}) + \boldsymbol{b}_{\text{enc}})) \\ \widehat{\boldsymbol{e}} &= \boldsymbol{W}_{\text{dec}}\boldsymbol{h}_{\text{aux}}, \end{aligned}$$

where the top-k$_{\text{aux}}(\cdot)$ operator selects the top k_{aux} dead features. Hence, the combined training objective becomes

$$\mathcal{L}(\boldsymbol{\theta}) = \mathcal{L}_{\text{reconstruct}}(\boldsymbol{\theta}) + \lambda\mathcal{L}_{\text{aux}}(\boldsymbol{\theta}) = \|\boldsymbol{v} - \widehat{\boldsymbol{v}}\|^2 + \lambda\|\boldsymbol{e} - \widehat{\boldsymbol{e}}_{\text{aux}}\|^2, \tag{3.1}$$

where $\boldsymbol{\theta} = \{\boldsymbol{W}_{\text{enc}}, \boldsymbol{W}_{\text{dec}}, \boldsymbol{b}_{\text{pre}}, \boldsymbol{b}_{\text{enc}}\}$ denotes the trainable parameters, and λ balances the loss contribution of the top-k active features and the top-k_{aux} dead features.

Chapter 4

Feature-Level Analysis

This chapter investigates the internal mechanisms of Stable Diffusion v2.1 through a feature-level analysis. We accomplish this by employing Sparse Autoencoders (SAEs) to decompose the activation vectors from key cross-attention blocks into sparse, interpretable dictionaries of features. We first detail the experimental setup for training these autoencoders and validate their performance. We then use the learned dictionaries to analyze the temporal dynamics of SAE features throughout the diffusion process. Finally, we conduct targeted interventions on the features to probe their functional roles and demonstrate their influence on the final generated image.

4.1 Experimental Setup

We generate the training data for our SAEs by extracting internal activations from the SD v2.1 model at an image resolution of 512×512. During image generation, we employ the Denoising Diffusion Implicit Models (DDIM) [43] scheduler with 25 denoising steps, a common choice that provides a favorable trade-off between image quality and computational cost. Using a 50,000-sample subset of the LAION-COCO [41] prompt dataset, we capture latent representations from the residual updates of two key cross-attention blocks (see Figure 3.1): the `down.2.0` block, previously identified as crucial for encoding objects and style [2], and the `up.1.1` block, which allows us to study representations within the model's upsampling pathway. Specifically, we extract these dense feature maps from the conditional forward pass of the classifier-free guidance mechanism, using a guidance scale of 9 to ensure strong prompt adherence. For a primer on (classifier-free) guidance in text-to-image diffusion models, we refer to Appendix A.2. This process yields a dense tensor of shape $16 \times 16 \times 1,280$ for each prompt at each of the 25 diffusion steps. By treating the vector at each of the $16 \times 16 = 256$ spatial locations as an independent training example, we compile a total training

dataset of 3.2×10^8 vectors, each with a (channel) dimension $d = 1280$.

Using this dataset, we follow Surkov et al. [46] and train TopK sparse autoencoders with a dictionary size of $n_f = 5120$ (a $4\times$ expansion factor relative to the channel dimension $d = 1280$) and $k = 10$ for an adequate tradeoff between reconstruction fidelity and sparsity. We employ two techniques to ensure training stability. First, to mitigate dead features, which we define as features that remain inactive over 1×10^7 training samples, we utilize the auxiliary loss from Equation 3.1 with $k_{aux} = 256$ and a weight of $\lambda = 1/32$ as in Gao et al. [13]. Second, to prevent the emergence of dominant features, we re-normalize the decoder's dictionary vectors to unit norm before each optimizer step. Moreover, we follow Bricken et al. [4] and initialize $\boldsymbol{b}_{pre}$ to be the geometric median of the training data. We train the SAEs for a total of 2×10^9 activation vectors using the Adam [24] optimizer ($\beta_1 = 0.9,\ \beta_2 = 0.999$) with a batch size of 4,096. We hold the learning rate constant at $\approx 3.578 \times 10^{-4}$, a value we derive from the scaling laws in Gao et al. [13].

4.2 Reconstruction Performance and Feature Dynamics

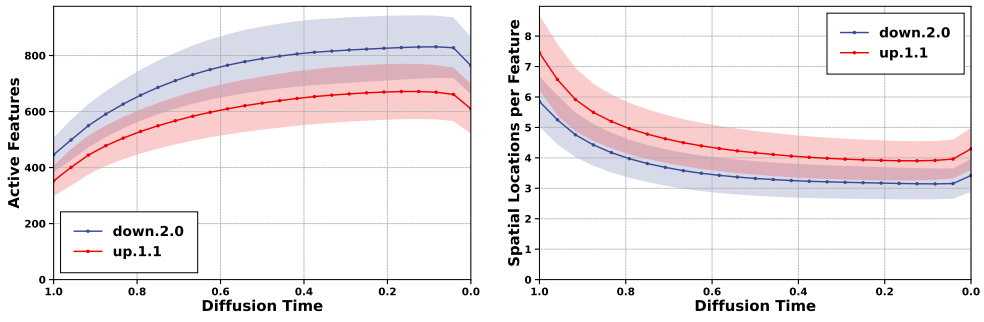
Our trained SAEs demonstrate high-fidelity performance both at the cross-attention update level and in downstream image generation, as detailed in Table 4.1. Internally, they achieve reasonably high R^2 scores without any dead features (features remaining inactive over 1×10^7 train samples), yielding reconstruction performance comparable to the SAEs from prior work on multi-step diffusion models [42, 49] that used similar n_f and k values. Crucially, when replacing the original cross-attention update with the SAE reconstruction during synthesis, the final images remain perceptually similar to those generated without the substitution. We evaluate this across 1,000 distinct test prompts from the LAION-COCO [41] dataset for each of five seeds. This perceptual similarity is evidenced by low Learned Perceptual Image Patch Similarity (LPIPS) [53] scores and small mean and median pixel-wise Manhattan distances, confirming that our learned dictionary preserves essential information for image generation. We define the latter metrics as the mean or median of the element-wise ℓ_1 norm between the two image tensors, with pixel values normalized to $[0, 1]$, taken across all pixels and channels. Moreover, the SAEs exhibit notably strong cross-resolution generalization capabilities. Despite being trained exclusively on data stemming from 512×512 images, the SAEs maintain similar reconstruction performance when applied to 768×768 image generation, and even show a slight improvement in some downstream quality metrics. Finally, we refer to Figure B.1 and Figure B.2 for some visual reconstruction examples.

Having established the quality of our SAE, we now analyze the temporal

dynamics of feature usage during the reverse diffusion process. Figure 4.1 reveals that during the early stage at high noise levels (time $t \in [0.6, 1.0]$), the number of active features is comparatively low, yet they are spatially diffuse and active across a broad set of pixels. This suggests that the initial denoising steps rely on a smaller, more foundational, and global set of concepts. As the process continues, a much wider array of features becomes active across fewer spatial locations during the mid-to-late stages. Finally, the active feature count declines as the process concludes, indicating a final refinement phase that requires a narrower set of features. Interestingly, the encoder block `down.2.0` consistently employs a larger number of features that are more spatially focused, which aligns with its hypothesized role in defining specific objects, compositional elements, and global style. In contrast, the decoder block `up.1.1` utilizes a smaller, more diffuse set of features.

Table 4.1: Reconstruction Performance of Sparse Autoencoders. Performance comparison of Sparse Autoencoders (SAEs) trained on two distinct cross-attention blocks. The table reports reconstruction quality using the fraction of variance explained (R^2), the percentage of dead features (as measured at the end of training), the Fréchet Inception Distance (FID), and the Learned Perceptual Image Patch Similarity (LPIPS). Additionally, it reports the mean pixel-wise Manhattan distance (Mean) and pixel-wise median Manhattan distance (Median) between the original and reconstructed images. Metrics are averaged over 1,000 distinct prompts, and standard deviations in parentheses are across five seeds.

Block	Resolution	R^2 (%) ↑	Dead Features (%) ↓	LPIPS ↓	Mean ↓	Median ↓
down.2.0	512×512	36.60 ($\pm$0.25)	0.00	0.492 ($\pm$0.018)	0.152 ($\pm$0.007)	0.100 ($\pm$0.003)
	768×768	36.48 ($\pm$0.44)	0.00	0.410 ($\pm$0.011)	0.104 ($\pm$0.007)	0.067 ($\pm$0.005)
up.1.1	512×512	39.43 ($\pm$0.41)	0.00	0.444 ($\pm$0.018)	0.136 ($\pm$0.006)	0.085 ($\pm$0.003)
	768×768	40.30 ($\pm$0.45)	0.00	0.361 ($\pm$0.011)	0.092 ($\pm$0.006)	0.055 ($\pm$0.004)



(a) Average number of distinct active features. **(b)** Average number of active spatial locations.

Figure 4.1: SAE Feature Activity Across Diffusion Time. **(a)** Average number of distinct active features and **(b)** average spatial locations per active feature across diffusion time for U-Net blocks `down.2.0` and `up.1.1`. Solid lines show the mean, and shaded areas represent one standard deviation, computed across 10,000 test prompts to show prompt-to-prompt variability.

4.3 Top Activating Images

To investigate the roles of individual features, we adapt the methodology of Surkov et al. [46] to the multi-step setting by quantifying the influence of a feature $\varphi \in [n_f]$ for a given input, defined by prompt $\boldsymbol{c}$ and initial noise $\boldsymbol{z}_{1.0}$, in a test dataset $\mathcal{D}_{\text{test}}$ and diffusion timestep $t \in [0, 1]$ by its mean spatial activation, that is

$$\mu_t^{\varphi}(\boldsymbol{c}, \boldsymbol{z}_{1.0}) = \frac{1}{hw} \sum_{i=1}^{h} \sum_{j=1}^{w} \boldsymbol{S}_{ij,t}^{\varphi}(\boldsymbol{c}, \boldsymbol{z}_t).$$

Furthermore, to summarize a feature's overall importance across the entire denoising process, we consider our set of $T = 25$ discrete timesteps $\mathcal{T} = \left\{\tau_i | \tau_i = 1 - \frac{i-1}{T-1},\ i \in [T]\right\}$ and define the average activation magnitude

$$\bar{\mu}^{\varphi}(\boldsymbol{c}, \boldsymbol{z}_{1.0}) = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \mu_t^{\varphi}(\boldsymbol{c}, \boldsymbol{z}_t),$$

as well as its peak activation time

$$t_{\text{peak}}^{\varphi}(\boldsymbol{c}, \boldsymbol{z}_{1.0}) = \arg\max_{t \in \mathcal{T}} \mu_t^{\varphi}(\boldsymbol{c}, \boldsymbol{z}_t).$$

This activation scoring enables us to identify the most representative image examples for any feature $\varphi \in [n_f]$. We compute these metrics for 5,000 input prompts over five different random seeds. We note that alternative aggregation strategies, such as taking the temporal maximum rather than the average, proved less effective at capturing representative feature behavior, as they overemphasize transient activation spikes rather than consistent behavior throughout the denoising process. We then visualize a feature by identifying the top nine dataset examples ranked by their average activation magnitude $\bar{\mu}^{\varphi}$. Additionally, we overlay a spatial heatmap of the sparse activations $\boldsymbol{S}_{t_{peak}^{\varphi}}^{\varphi}$ at their peak activation time.

As Figure 4.2 demonstrates, this analysis reveals a diverse range of learned features in the SAE for the `down.2.0` block. We identify features that are selective for specific objects and others that correspond to broader artistic styles. For example, feature 3094 and feature 2165 consistently activate on roses and dog faces, respectively, whereas feature 2420 corresponds to an anime art style. Beyond these and numerous other features tied to specific semantic concepts, we identify various others that govern more general visual properties, such as texture and composition. While their visual effect is often apparent, such as the generation of grid-like patterns, they are usually not highly selective to a particular object or style. Overall, the types of features we identify appear to be analogous to the feature taxonomies discovered in other recent SAE-based investigations of Stable Diffusion models [6, 46, 49].

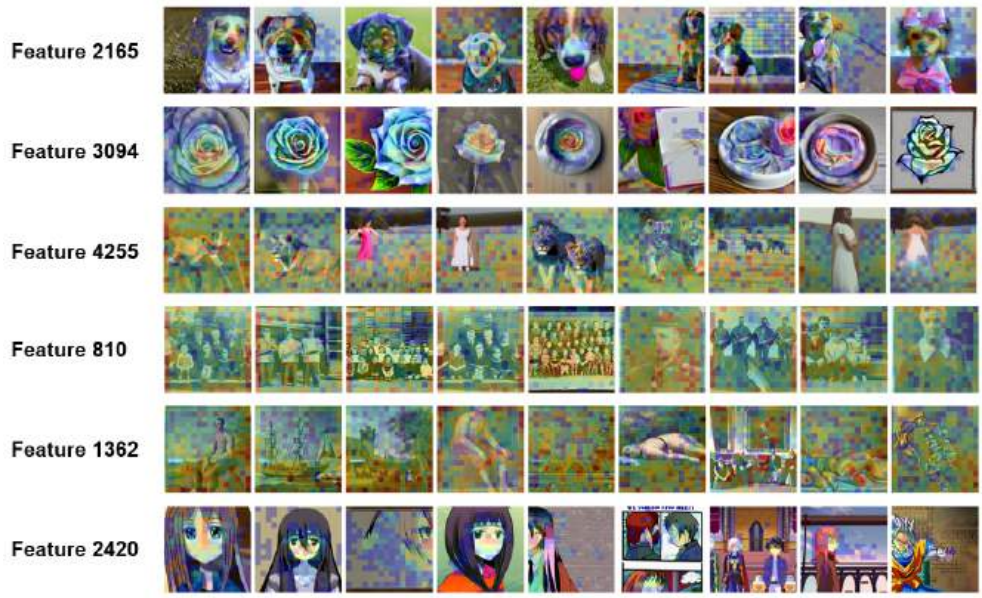


Figure 4.2: Top-Activating Images for down.2.0 SAE Features. Each row displays a feature's top nine most-activating examples from a test set, consisting of 5,000 LAION-COCO prompts across five different seeds, that maximally activate a specific feature, as measured by the average activation $\bar{\mu}^{\varphi}$. The heatmaps overlaid on the images visualize the sparse activations $S^{\varphi}_{t^{\varphi}_{peak}}$ at the feature's peak activation time, with warmer colors indicating higher values.

4.4 Targeted Feature Interventions

We investigate the functional role of learned features by performing targeted interventions. For this, we adopt the intervention framework from Surkov et al. [46]. To effectively steer the generation in a model using classifier-free guidance, we modulate the strength of a feature φ by adding it to the text-conditioned forward pass and subtracting it from the unconditional pass. This approach effectively isolates a feature's influence, as shown in Figure B.4 and is consistent with the interventions for the SDXL base model [35] in Surkov et al. [46]. This intervention is applied to the update tensor $\Delta \boldsymbol{D}_t \in \mathbb{R}^{h \times w \times d}$ as follows:

$$\Delta \boldsymbol{D}'_t = \begin{cases} \Delta \boldsymbol{D}_t + \xi \left(\mathbf{1}_{h,w} \otimes \boldsymbol{f}_{\varphi} \right) & \text{for the conditional pass} \\ \Delta \boldsymbol{D}_t - \xi \left(\mathbf{1}_{h,w} \otimes \boldsymbol{f}_{\varphi} \right) & \text{for the unconditional pass} \end{cases} \tag{4.1}$$

where $\xi \in \mathbb{R}$ is the intervention strength, $\boldsymbol{f}_{\varphi} \in \mathbb{R}^d$ is the learned feature vector, and the outer product $\otimes$ with a matrix of ones $\mathbf{1}_{h,w} \in \mathbb{R}^{h \times w}$ broadcasts the feature, applying the same vector $\boldsymbol{f}_{\varphi}$ uniformly to all spatial locations. The tensors $\Delta \boldsymbol{D}_t$ and $\Delta \boldsymbol{D}'_t$ represent the original and modified cross-attention block updates, respectively.

We apply this global intervention to the features from Figure 4.2. The results, shown in Figure 4.3, are highly consistent with the roles we inferred from the top-activating images and support our earlier hypothesis that features 2165, 3094, and 4255 are responsible for object knowledge. Features 810, 1362, and 2420 appear to govern overall style. Specifically, amplifying object-centric features causes the Roman emperor to be replaced by a human face, a dog, or a field of grass, respectively, and amplifying style-centric features imposes an old-school photography, painterly, or anime aesthetic onto the portrait without removing the subject itself. This observation is in line with Basu et al. [2], who also find `down.2.0` in SD v2.1 to be highly relevant for style and object generation. In contrast, interventions on `up.1.1` features reveal a different function. The block's use of a smaller, more diffuse feature set (as shown in Figure 4.1) suggests a role in rendering and texturing, rather than composition. Our intervention experiments for `up.1.1` confirm this by yielding only subtle textural changes compared to the conceptual shifts seen in `down.2.0`. Finally, our experiments reveal a key generalization capability, as the image-level impact of SAE feature interventions appears to be consistent across image resolutions (cf. Figure B.3).

These intervention results align with recent work [46, 42, 6] demonstrating that amplifying SAE features can substantially and controllably alter both object identity and artistic style within specific cross-attention layers. Notably, feature injections in modern distilled diffusion models, such as those examined by Surkov et al., appear to yield qualitatively more artistic effects. Moreover, SAE feature intervention examples in SD v2.1 appear qualitatively more salient than those shown by Tinaz et al. [49] for SD v1.4. However, this visual difference could stem from the use of different Stable Diffusion models or variations in the implementation details of the intervention.

To investigate how the influence of features evolves during the generation process, we stratify our interventions across three distinct windows: early-stage ($t \in [0.6, 1.0]$), mid-stage ($t \in [0.2, 0.6]$), and late-stage ($t \in [0.0, 0.2]$). Interventions during the initial timesteps (Figure 4.4) have a profound and global impact on the final image output, capable of altogether redefining the image's core subject and style, yielding results similar to those in Figure 4.3 with continuous interventions at all timesteps. Specifically, amplifying object features at this early stage replaces the Roman emperor with an entirely different subject, such as a dog (feature 2165). However, we note that for feature 4255, intervening only during the early stage is insufficient to fully transform the emperor into grass. Crucially, this stage is also critical for establishing the global aesthetic as activating style features transforms the entire image into an old photograph (feature 810), a medieval painting (feature 1362), or an anime drawing (feature 2420).

As the generation progresses into the mid-stage (Figure 4.5), the founda-

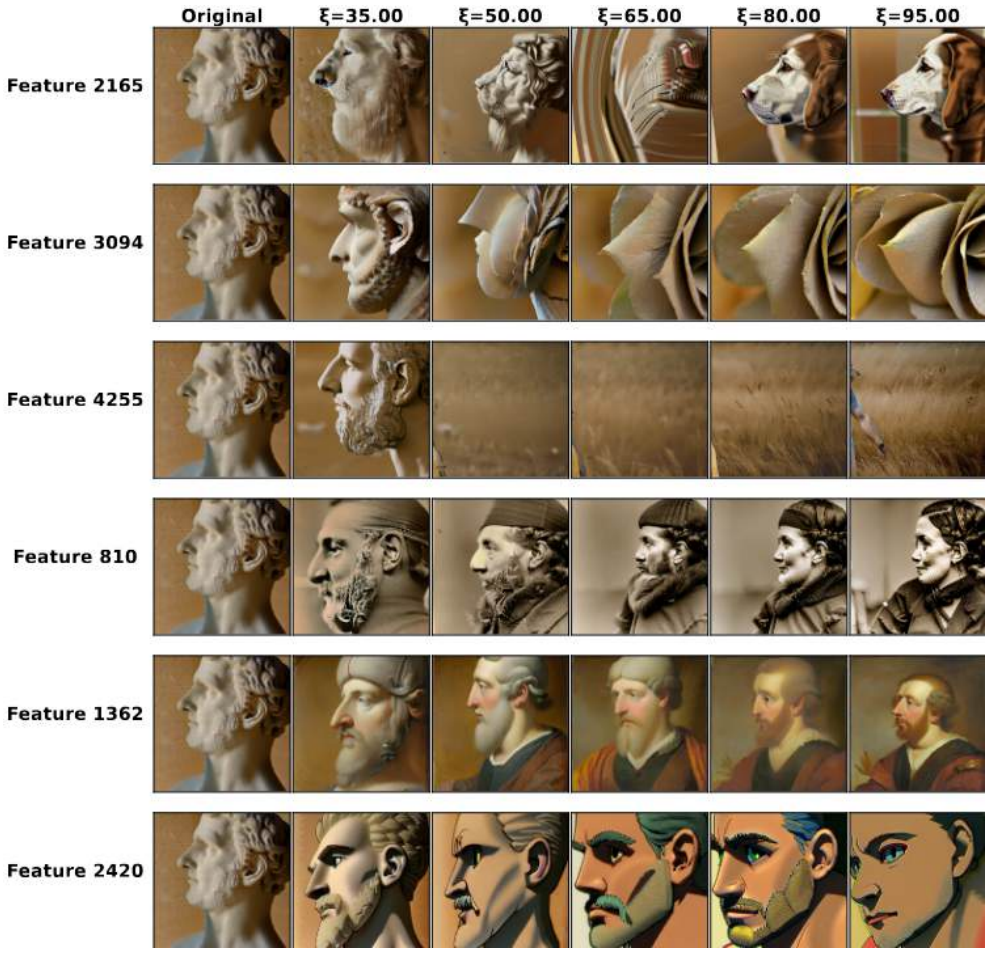


Figure 4.3: Steering Image Generation by Amplifying Individual Features. All images are generated using the prompt "Portrait of a stoic Roman emperor, profile view.". Each row demonstrates the effect of amplifying a single feature φ at every timestep throughout the generation process. The leftmost column shows the baseline generation without intervention. In the subsequent columns, the feature is amplified by adding $\xi \cdot f_\varphi$ to the cross-attention block's residual update, with strength ξ increasing from 35 to 95.

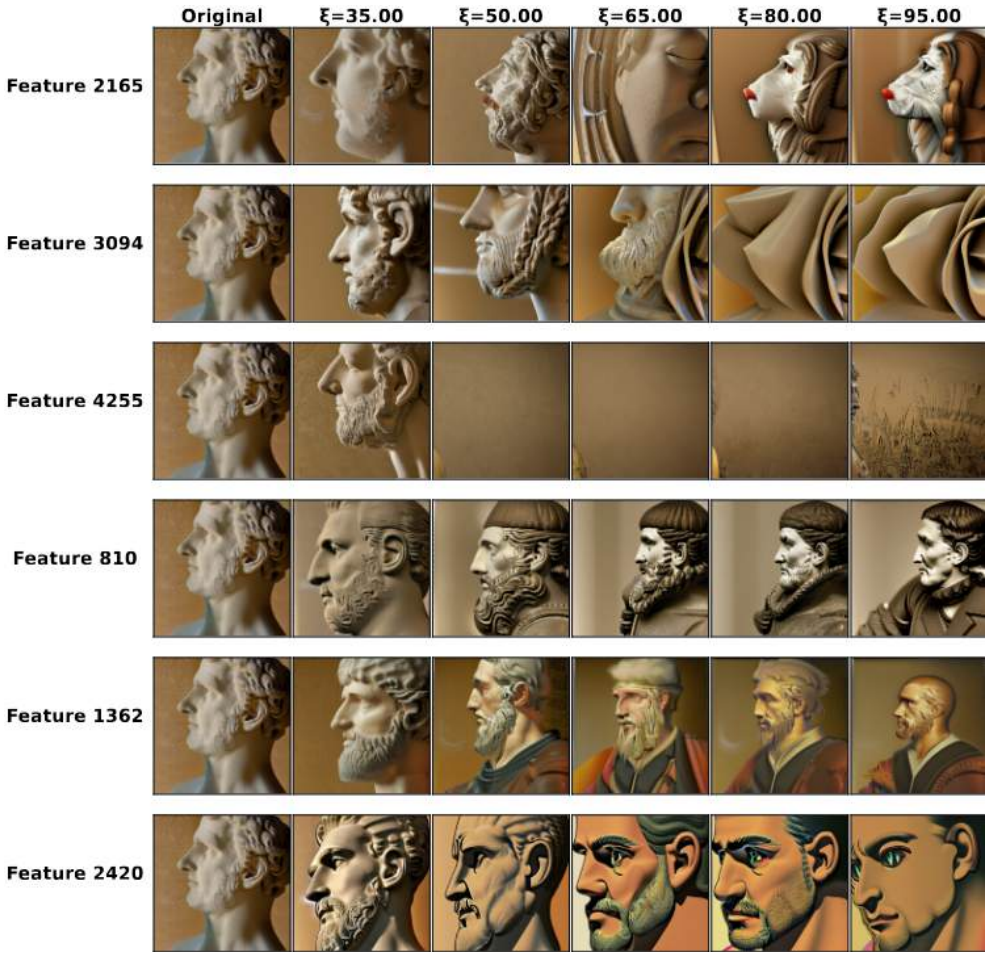


Figure 4.4: Steering Image Generation with Early-Stage Feature Interventions. All images are generated using the prompt "Portrait of a stoic Roman emperor, profile view.". Each row demonstrates the effect of amplifying a single feature φ during the initial timesteps ($t \in [0.6, 1.0]$). The leftmost column shows the baseline generation without intervention. In the subsequent columns, the feature is amplified by adding $\xi \cdot f_\varphi$ to the `down.2.0` block's residual update, with strength ξ increasing from 35 to 95.

tional structure is partially formed, causing concept features to blend with the existing composition rather than replace it. For instance, activating the "dog" feature (2165) no longer creates a distinct dog; instead, it merges canine facial features with the emperor's established profile. Similarly, the style features refine the existing structure by applying stylistic modifications rather than determining the initial aesthetic.

Figure 4.5: Steering Image Generation with Mid-Stage Feature Interventions. All images are generated using the prompt "Portrait of a stoic Roman emperor, profile view.". Each row demonstrates the effect of amplifying a single feature φ during the intermediate timesteps ($t \in [0.2, 0.6]$). The leftmost column shows the baseline generation without intervention. In the subsequent columns, the feature is amplified by adding $\xi \cdot f_\varphi$ to the `down.2.0` block's residual update, with strength ξ increasing from 35 to 95.

In the final stages of generation, the image's content and composition are largely fixed. As a result, the visible impact of feature interventions diminishes considerably (Figure 4.6). Activating any of the features under consideration induces only subtle textural refinements and high-frequency modi-

fications. Specifically, for our selected subset of features, the interventions introduce minor artifacts and slight variations in surface texture, leaving the emperor's portrait fundamentally unchanged.

Figure 4.6: Steering Image Generation with Late-Stage Feature Interventions. All images are generated using the prompt "Portrait of a stoic Roman emperor, profile view.". Each row demonstrates the effect of amplifying a single feature φ during the final timesteps ($t \in [0.0, 0.2]$). The leftmost column shows the baseline generation without intervention. In the subsequent columns, the feature is amplified by adding $\xi \cdot f_\varphi$ to the `down.2.0` block's residual update, with strength ξ increasing from 35 to 95.

Finally, we systematically quantify the collective impact of our learned feature basis across the generation process. For this, we utilize an aggregate approach we term a knockout intervention, as our primary interest is to measure the collective influence of the entire dictionary at different stages. Furthermore, the impact of any single feature can be marginal and is often highly context- and time-dependent, making a one-by-one analysis less informative for our goals.

Let $\Phi_t = \{\varphi \in [n_f] \mid \|S_t^\varphi\|_0 > 0\}$ be the set of features with non-zero sparse activation maps at timestep t. Our knockout intervention modifies the cross-attention update tensor $\Delta D_t \in \mathbb{R}^{h \times w \times d}$ by subtracting the aggregated feature representations during the conditional pass and adding them during the unconditional pass, that is,

$$\Delta D_t' = \begin{cases} \Delta D_t - \sum_{\varphi \in \Phi_t} \left(S_t^\varphi \otimes f_\varphi \right) & \text{for the conditional pass} \\ \Delta D_t + \sum_{\varphi \in \Phi_t} \left(S_t^\varphi \otimes f_\varphi \right) & \text{for the unconditional pass} \end{cases}$$

Here, $S_t^\varphi \in \mathbb{R}^{h \times w}$ is the sparse activation map for feature φ at timestep t, $f_\varphi \in \mathbb{R}^d$ is the feature vector, and the outer product $\otimes$ broadcasts the feature vector, effectively scaling it by the corresponding sparse activation value $S_{ij,t}^\varphi$ at each spatial location (i, j).

We apply this knockout intervention at various stages across 1,000 distinct input prompts over five different random seeds. Table 4.2 presents the results, quantifying the image-level impact of these interventions within the `down.2.0` and `up.1.1` cross-attention blocks. The findings reveal a strong temporal dependency that is consistent across all metrics and coincides with our previous visual investigations. Namely, interventions during the early stage substantially change the final image output; this impact diminishes considerably during the mid-stage and becomes markedly smaller in the late stage. This trend holds for both the `down.2.0` and `up.1.1` blocks, suggesting the phenomenon is not localized to a specific part of the U-Net architecture.

Table 4.2: Impact of Knockout Interventions at Different Stages. The change induced by the intervention on the final generated image is measured using LPIPS and the mean and median pixel-wise Manhattan distances. Interventions are applied at different stages: Early-Stage ($t \in [0.6, 1.0]$), Mid-Stage ($t \in [0.2, 0.6]$), and Late-Stage ($t \in [0.0, 0.2]$). Higher metric values indicate a greater impact of the intervention. Results are averaged across 1,000 distinct prompts, where each prompt was generated using five different random seeds. Standard deviations across the five random seeds are reported in parentheses.

Block	Intervention Stage	LPIPS	Mean	Median
down.2.0	Early-Stage	0.580 ($\pm$0.015)	0.187 ($\pm$0.009)	0.131 ($\pm$0.004)
	Mid-Stage	0.272 ($\pm$0.016)	0.079 ($\pm$0.002)	0.044 ($\pm$0.002)
	Late-Stage	0.105 ($\pm$0.014)	0.038 ($\pm$0.001)	0.022 ($\pm$0.001)
up.1.1	Early-Stage	0.554 ($\pm$0.015)	0.171 ($\pm$0.010)	0.116 ($\pm$0.006)
	Mid-Stage	0.288 ($\pm$0.015)	0.082 ($\pm$0.003)	0.046 ($\pm$0.001)
	Late-Stage	0.101 ($\pm$0.008)	0.039 ($\pm$0.002)	0.022 ($\pm$0.001)

Chapter 5

Circuit-Level Analysis

While sparse dictionary learning identifies an interpretable vocabulary of features, a deeper mechanistic understanding requires identifying the computational circuits that compose them. Complex visual concepts emerge not from isolated features, but from their structured interactions. This chapter extends our analysis from single-feature interventions to the discovery and validation of computational subgraphs, or circuits, that govern concept formation within the diffusion process. We begin by formalizing activation patching, a technique for transplanting sets of feature activations between generation processes. We then adapt methodologies from causal mediation analysis [34] and interpretability research in large language models [31] to identify and qualitatively evaluate these sparse feature circuits in a diffusion context.

5.1 Concept Transfer via Activation Patching

To investigate how features act in concert, we extend our analysis from single-feature *do*-interventions to patching entire sets of features. Activation patching is a causal intervention technique that probes the function of neural network components by transplanting their activations from a "source" context to a "destination" context [16, 32, 51]. Following Surkov et al. [46], we apply activation patching to SAE-decomposed features using differential scoring to identify transferable feature sets. Critically, we extend their approach to multi-step diffusion models with classifier-free guidance. Specifically, we alter the generation of an image from a destination prompt $\boldsymbol{c}_{\text{dest}}$ by replacing a subset of its feature activations with those derived from a source prompt $\boldsymbol{c}_{\text{src}}$, thereby assessing the causal impact of specific feature sets on the final generated image.

A successful transfer requires identifying features that are most characteristic of the source concept relative to the destination concept. To do this, we

adapt the differential analysis method from Surkov et al. [46]. Let $\bar{\mu}_\varphi(\boldsymbol{c})$ be the mean activation of a feature φ for a given prompt $\boldsymbol{c}$ across all spatial locations and diffusion timesteps. We found the temporal mean to be most effective; while we also experimented with temporal maximum aggregation, it tended to identify less transferable, global-structure features. We compute a differential score γ_φ for each feature φ:

$$\gamma_\varphi = \frac{\bar{\mu}_\varphi(\boldsymbol{c}_{src})}{\sum_{\varphi'} \bar{\mu}_{\varphi'}(\boldsymbol{c}_{src})} - \frac{\bar{\mu}_\varphi(\boldsymbol{c}_{dest})}{\sum_{\varphi'} \bar{\mu}_{\varphi'}(\boldsymbol{c}_{dest})}.$$

A high positive score γ_φ indicates that feature φ is disproportionately more active for the source prompt, making it a candidate for transfer. We select the top-m features with the highest positive scores as the source set $\mathcal{K}_{\text{src}}$ and the top-m features with the most negative scores as the destination set $\mathcal{K}_{\text{dest}}$:

$$\mathcal{K}_{\text{src}} = \arg\text{top-m}_\varphi(\gamma_\varphi), \quad \mathcal{K}_{\text{dest}} = \arg\text{top-m}_\varphi(-\gamma_\varphi). \tag{5.1}$$

Once these feature sets are identified, we construct a patch vector, $\boldsymbol{P}_t(\boldsymbol{c}, \mathcal{K})$, by summing the contributions of all features φ within a given set $\mathcal{K}$. This reconstruction is calculated by scaling each feature $\boldsymbol{f}_\varphi$ by its correspoinding spatial activation map $\boldsymbol{S}_t^\varphi(\boldsymbol{c})$:

$$\boldsymbol{P}_t(\boldsymbol{c}, \mathcal{K}) = \sum_{\varphi \in \mathcal{K}} (\boldsymbol{S}_t^\varphi(\boldsymbol{c}) \otimes \boldsymbol{f}_\varphi).$$

During the generation of the destination prompt, we modify the residual update tensor at each timestep $t \in [0, 1]$. Thereby, the core idea is first to remove the features most characteristic of the destination concept and then insert the features most characteristic of the source concept. Again, as demonstrated in Figure B.4, we find it most effective to apply the intervention differentially to the conditional and unconditional forward passes. Hence, the modified residual update $\boldsymbol{\Delta D}_t'$ is calculated as

$$\Delta \boldsymbol{D}_t' = \begin{cases} \Delta \boldsymbol{D}_t(\boldsymbol{c}_{dest}) - \alpha \cdot \boldsymbol{P}_t(\boldsymbol{c}_{dest}, \mathcal{K}_{dest}) + \beta \cdot \boldsymbol{P}_t(\boldsymbol{c}_{src}, \mathcal{K}_{src}) & \text{for the conditional pass,} \\ \Delta \boldsymbol{D}_t(\boldsymbol{c}_{dest}) + \alpha \cdot \boldsymbol{P}_t(\boldsymbol{c}_{dest}, \mathcal{K}_{dest}) - \beta \cdot \boldsymbol{P}_t(\boldsymbol{c}_{src}, \mathcal{K}_{src}) & \text{for the unconditional pass,} \end{cases} \tag{5.2}$$

where the scaling factors $\alpha, \beta \in \mathbb{R}_+$ control the strength of the patch subtraction and addition. The most direct intervention is to add the source patch to the destination's residual stream by setting $\alpha = 0$ and $\beta = 1$. This, however, often leads to interference as we intervene on a single cross-attention block only, whose signal likely gets diluted by other cross-attention blocks

on which we do not intervene. To avoid this, based on empirical results, we set $\alpha = 1$ and $\beta = 2$ for all experiments to produce salient visual changes.

We demonstrate the efficacy of this methodology through a series of concept transfer experiments for `down.2.0`, shown in 5.1. The figure displays a grid of activation patching experiments where columns represent different values of m, where m source and m patch features are affected by the intervention. Notably, our method achieves robust semantic transfer with a small set of features, often as few as $m = 5$ to $m = 20$ features. In contrast, the spatially masked patching method of Surkov et al. [46] typically requires a far larger set of features for successful transfer. We hypothesize this significant gain in efficiency stems from our approach of transferring a feature's complete spatial activation map globally. The method of Surkov et al. [46], by contrast, not only constrains features to a semantic mask but also collapses the activation map within that mask to a single statistical value, which is then broadcast across the target region, discarding the feature's native spatial structure.

Again, as in the single-feature interventions, we find that this behaviour is consistent across different resolutions. Specifically, Figure B.5 showcases that the activation patching experiments translate to a higher image resolution of 768×768 despite the SAE being trained on a lower resolution of 512×512. Moreover, we note that this phenomenon also holds for the `up.1.1` block, as is illustrated in Appendix Figure B.6 for a resolution of 512×512 and Figure B.7 for 768×768. However, we observe that `down.2.0` tends to require fewer SAE features for high-quality concept transfers, which we attribute to its role as an object knowledge block.

Finally, Figure 5.2 shows the activation patching results stratified by diffusion stage for two selected comparative prompt pairs. This figure reveals that, also when intervening on multiple features at once, there is a clear temporal hierarchy in concept emergence, where interventions are most powerful in the early stages of generation and their impact diminishes significantly over time. In the early stage ($t \in [0.6, 1.0]$), patching has a salient global effect that fundamentally alters an image's core subject and composition. This is evident where "cat" features immediately transform the bird's entire form and "forest" features completely redefine the bear's background. This global influence wanes in the mid-stage ($t \in [0.2, 0.6]$), where interventions primarily refine the established structure or create hybrid concepts rather than completely overwrite it. Finally, activation patching applied to the late stage ($t \in [0.0, 0.2]$) has a negligible effect, as the model has already committed to the image's primary compositions.

Figure 5.1: Feature Activation Transfer in down.2.0. Activation patching is performed in the `down.2.0` cross-attention block across all diffusion timesteps. The transfer is achieved by subtracting the top-m features most characteristic of the control concept (with a scaling factor of $\alpha = 1$) and adding the top-m features most characteristic of the source concept (with a scaling factor of $\beta = 2$). Columns show the effect of transplanting an increasing number of feature activations ($m \in \{1, 5, 20, 50, 200\}$) from a source to a control concept. From top to bottom, the rows demonstrate the following conceptual transfers: "forest" in place of "arctic"; "cat" in place of "bird"; "dog" in place of "bird"; "mountain lake" in place of "ocean bay"; "cathedral" in place of "glass skyscraper"; "calm ocean" in place of "rough ocean".

Figure 5.2: Stratified Feature Activation Transfer in down.2.0. To analyze the temporal emergence of concepts, activation patching is performed in the `down.2.0` cross-attention block in a stratified manner. The transfer, achieved by subtracting top-m control features ($\alpha = 1$) and adding top-m source features ($\beta = 2$), is confined to one of three distinct diffusion stages: early ($t \in [0.6, 1.0]$), middle ($t \in [0.2, 0.6]$), or late ($t \in [0.0, 0.2]$). The columns show the effect of increasing the number of transferred features m. The rows are grouped by these intervention stages, demonstrating the varying impact of timing on conceptual transfers like "cat" in place of "bird" and a "forest" background in place of an "arctic" one.

5.2 A Framework for Temporal Circuit Discovery

Activation patching demonstrates the collective power of feature sets but does not reveal the underlying computational mechanisms. To understand how concepts are formed and refined over time, we must analyze the causal pathways and interactions between features across different timesteps. This leads us to the discovery of computational circuits.

A key insight from Marks et al. [31] is that any analysis restricted solely to the learned SAE features is necessarily incomplete. The SAE decomposition is approximate, and the reconstruction error, $\boldsymbol{e}_t = \boldsymbol{\Delta D}_t - \widehat{\boldsymbol{\Delta D}}_t$, contains causally pivotal information not captured by the dictionary. Discarding this

error term would mean ignoring a potentially crucial part of the model's computation. Therefore, in the following, we treat the SAE decomposition not as an approximation but as a complete replacement of the residual cross-attention update vector, that is,

$$\Delta \boldsymbol{D}_t = \underbrace{\mathrm{DEC}(\mathrm{ENC}(\Delta \boldsymbol{D}_t))}_{\text{reconstruction} \quad \widehat{\Delta \boldsymbol{D}_t}} + \boldsymbol{e}_t = \sum_{\varphi=1}^{n_f} \left(\boldsymbol{S}_t^{\varphi} \otimes \boldsymbol{f}_{\varphi}\right) + \boldsymbol{b}_{pre} + \boldsymbol{e}_t,$$

where $\otimes$ is the outer product that applies the feature vectors to all spatial locations. This principled decomposition allows us to analyze the model's behavior in terms of both interpretable sparse features and an uninterpreted, but causally potent, error component.

While the diffusion process is defined over a continuous time variable $t \in [0, 1]$, numerical solvers operate on a discrete schedule. To facilitate the definition of a circuit, we again define the set of discrete timesteps under consideration as $\mathcal{T} = \{\tau_i \mid \tau_i = 1 - \frac{i-1}{T-1},\ i \in [T]\}$, where in all of our experiments $T = 25$. Hence, in the following, we denote with $\tau_1 = 1.0$ the earliest, pure noise timestep and with τ_T the final timestep. We now formally define a temporal sparse feature circuit $\mathcal{C}$ as a weighted, directed acyclic graph (DAG), $\mathcal{C} = (\mathcal{V}, \mathcal{E}, W)$ over these discrete timesteps:

- **Vertices** $v \in \mathcal{V}$: Each vertex represents a specific computational component at a specific timestep. A vertex is a tuple $v = (\kappa, \tau_i)$, where $\tau_i \in \mathcal{T}$ is a discrete timestep and the component κ is either an SAE feature index $\varphi \in [n_f]$ or a symbol `res` representing the SAE error term. Thus,
$$\mathcal{V} \subseteq \left(\{1, \ldots, n_f\} \cup \{\text{res}\}\right) \times \mathcal{T}.$$

- **Directed Edges** $e \in \mathcal{E}$: A directed edge $e = (v_i, v_j) \in \mathcal{V} \times \mathcal{V}$, with $v_i = (\kappa_i, \tau_i)$ and $v_j = (\kappa_j, \tau_j)$, exists if component κ_i at timestep τ_i causally affects the component κ_j at a subsequent timestep $\tau_j < \tau_i$ in the generation process. Hence,
$$\mathcal{E} \subseteq \{(v_i, v_j) \in \mathcal{V} \times \mathcal{V} \mid \tau_i > \tau_j, \quad \tau_i, \tau_j \in \mathcal{T}\}.$$

- **Weights** W: A function $W : \mathcal{E} \to \mathbb{R}$, $e \mapsto W(e)$ that maps each edge $e \in \mathcal{E}$ to a scalar weight, $W(e)$, quantifying the strength of the causal influence.

Having formalized circuits as computational subgraphs, we now turn to their discovery. The key challenge is identifying which features at which timesteps causally affect the model's output. For this, we adapt the sparse feature circuit discovery framework from Marks et al. [31] to the temporal

structure of diffusion models. In this context, we measure the total causal effect [34] of an early-stage intervention on a downstream metric $\mathcal{M}(\cdot)$. This is often termed an "indirect effect" in mechanistic interpretability [31], as the influence is mediated by all subsequent computational steps of the model, as is illustrated in Figure 5.3.

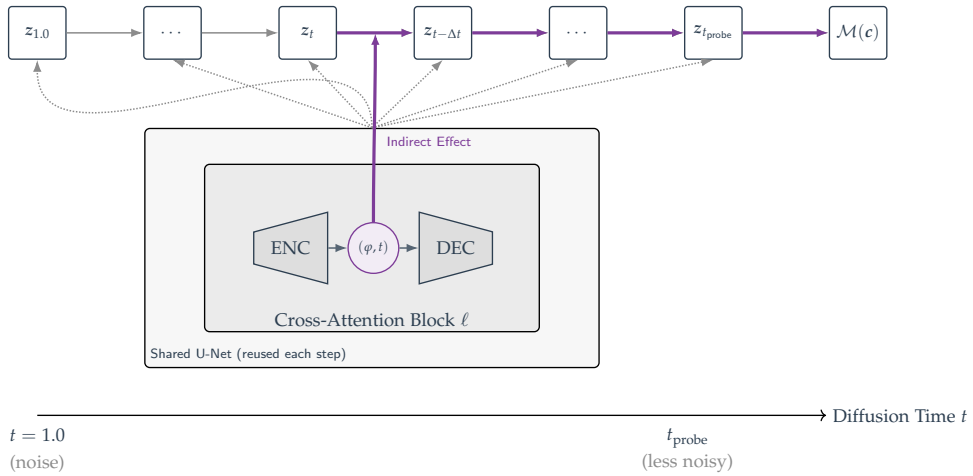


Figure 5.3: Measuring the Indirect Effect of a SAE Feature Component. The diagram illustrates a causal intervention to measure the influence of a single SAE feature component, (φ, t). At diffusion timestep t, we intervene on the activation of feature φ within cross-attention block ℓ. This causally alters the computation of the subsequent latent state, $z_{t-\Delta t}$. Because the U-Net is applied recurrently, this initial perturbation propagates through all subsequent denoising steps. The indirect effect of the component is then quantified by the resulting change in a diagnostic probe metric, $\mathcal{M}(c)$, evaluated on the cross-attention block output at a later timestep, t_{probe}.

For a node $v = (\kappa, \tau_i) \in \mathcal{V}$, its state is represented by a spatial activation pattern $\boldsymbol{a}_v = [\boldsymbol{a}_v^u; \boldsymbol{a}_v^c] \in \mathbb{R}^{2\times h\times w}$, comprising separate maps for the unconditional ($\boldsymbol{a}_v^u$) and conditional ($\boldsymbol{a}_v^c$) passes of classifier-free guidance. The component κ specifies whether this activation pattern corresponds to a sparse feature map $\boldsymbol{S}_t^\kappa$ if κ indexes a SAE feature, or a channel of the error tensor if $\kappa =$`res`. Hence, we can define the indirect effect of component v on metric $\mathcal{M}$ as

$$\text{IE}(\mathcal{M}; \boldsymbol{a}_v; \boldsymbol{c}_{dest}, \boldsymbol{c}_{src}) = \mathcal{M}(\boldsymbol{c}_{dest}|do(\boldsymbol{a}_v = \boldsymbol{a}_v^{src}))) - \mathcal{M}(\boldsymbol{c}_{dest}).$$

The expression $\mathcal{M}(\boldsymbol{c}_{dest}|do(\cdot))$ represents a counterfactual forward pass where the model is run with the $\boldsymbol{c}_{dest}$ input, but at timestep t, the representation $\boldsymbol{a}_t$ is intervened on and forcibly set to the value it would have taken on the $\boldsymbol{c}_{src}$ input. However, please note that interventions for circuit discovery differ from the activation patching explained in Section 5.1. Here, we perform a complete replacement of a component's state (both SAE features and residual) from a source context to a destination context. This is an analytical

intervention designed to measure causal effects, rather than a generative one designed to steer image generation.

Computing the exact IE for every component using this new form of activation patching is computationally infeasible, as it would require a separate forward pass for each of the thousands of components at every timestep under consideration. We therefore turn to efficient linear approximations. First, attribution patching [33, 27], which allows us to compute the indirect effect for all components using only two forward and a single backward pass through a first-order Taylor approximation to activation patching:

$$\widehat{\mathrm{IE}}_{\mathrm{atp}}(\mathcal{M}; \boldsymbol{a}_v; \boldsymbol{c}_{dest}, \boldsymbol{c}_{src}) = \left\langle \underbrace{\nabla_{\boldsymbol{a}_v} \mathcal{M}|_{\boldsymbol{a}_v = \boldsymbol{a}_v^{dest}}}_{\boldsymbol{g}_v}, \underbrace{\boldsymbol{a}_v^{src} - \boldsymbol{a}_v^{dest}}_{\boldsymbol{\delta}_v} \right\rangle_F ,$$

where the Frobenius inner product $\langle \cdot, \cdot \rangle_F$ sums the element-wise product of the two tensors. This aggregates the causal effects across all spatial positions (dimensions h and w) and sums the contributions from both the unconditional and conditional passes

$$\langle \boldsymbol{g}, \boldsymbol{\delta} \rangle_F = \sum_{b=1}^{2} \sum_{i=1}^{h} \sum_{j=1}^{w} \boldsymbol{g}_{b,i,j} \boldsymbol{\delta}_{b,i,j}.$$

However, Marks et al. [31] find that attribution patching underestimates the ground truth IE at early layers and that using integrated gradients [45] instead yields significantly more accurate estimates for these layers. In our setup, this increased accuracy is crucial for reliably discovering the long-range temporal dependencies that define circuits in diffusion models. Hence, we follow Marks et al. [31] and estimate the IE using

$$\begin{aligned} \widehat{\mathrm{IE}}_{\mathrm{ig}}(\mathcal{M}; \boldsymbol{a}_v; \boldsymbol{c}_{dest}, \boldsymbol{c}_{src}) &= \left\langle \int_0^1 \nabla_{\boldsymbol{a}_v} \mathcal{M}|_{\boldsymbol{a}_v = \alpha \boldsymbol{a}_v^{dest} + (1-\alpha)\boldsymbol{a}_v^{src}} d\alpha, \boldsymbol{a}_v^{src} - \boldsymbol{a}_v^{dest} \right\rangle_F \\ &\approx \left\langle \frac{1}{N} \sum_{i=0}^{N-1} \nabla_{\boldsymbol{a}_v} \mathcal{M}|_{\boldsymbol{a}_v = \frac{i}{N} \boldsymbol{a}_v^{dest} + \left(1 - \frac{i}{N}\right) \boldsymbol{a}_v^{src}}, \boldsymbol{a}_v^{src} - \boldsymbol{a}_v^{dest} \right\rangle_F , \end{aligned}$$

where in accordance with Marks et al. [31] we set $N = 10$. Finally, note that in case no comparative pairs are available, one can also perform zero ablation [31] to find components that are necessary for the formation of a specific concept, by setting $\boldsymbol{a}_v^{src} = \mathbf{0}$ in the above estimation procedures. This, however, comes at the cost of less precise analysis, with interventions that are often out of distribution [33].

Beyond identifying important nodes, we compute edge weights representing indirect effects between components across timesteps. We employ a Markov

chain assumption, restricting edge computation to consecutive timesteps (τ_i, τ_{i+1}). This approach captures the primary causal pathways and ensures computational tractability, directly mirroring the sequential nature of the denoising process. Concretely, we adapt the implementation of Marks et al. [31] and use a linear approximation through Jacobian-vector products (JVP) to compute the weight of an edge (v_i, v_j), where $v_i = (\kappa_i, \tau_i)$ and $v_j = (\kappa_j, \tau_j)$ with $j = i + 1$, that is,

$$W((v_i, v_j)) = \left\langle \boldsymbol{g}_{v_j}, \frac{\partial \boldsymbol{a}_{v_j}}{\partial \boldsymbol{a}_{v_i}} \boldsymbol{\delta}_{v_i} \right\rangle_F$$

where $\boldsymbol{g}_{v_j}$ is the (integrated) gradient of the metric $\mathcal{M}$ with respect to the late activations $\boldsymbol{a}_{v_j}$, and $\boldsymbol{\delta}_{v_i} = \boldsymbol{a}_{v_i}^{src} - \boldsymbol{a}_{v_i}^{dest}$ is the early activation difference. While the term $\frac{\partial \boldsymbol{a}_{v_j}}{\partial \boldsymbol{a}_{v_i}} \boldsymbol{\delta}_{v_i}$ is a Jacobian-vector product (JVP), the entire expression is most efficiently computed using a vector-Jacobian product (VJP). By re-associating the terms as

$$W((v_i, v_j)) = \left\langle \left(\frac{\partial \boldsymbol{a}_{v_j}}{\partial \boldsymbol{a}_{v_i}} \right)^T \boldsymbol{g}_{v_j}, \boldsymbol{\delta}_{v_i} \right\rangle_F,$$

we can calculate the VJP term $\left(\frac{\partial \boldsymbol{a}_{v_j}}{\partial \boldsymbol{a}_{v_i}} \right)^T \boldsymbol{g}_{v_j}$ with a single backward pass starting from the late component v_j. This makes the calculation tractable when assessing the influence of many early components at τ_i on a single late component of interest at $\tau_j = \tau_{i+1}$.

Of course, any gradient-based attribution requires a differentiable scalar metric $\mathcal{M}(\cdot)$ to serve as the objective. Since diffusion models lack discrete output vocabularies, we cannot directly apply the logit difference metric commonly used in LLM mechanistic interpretability literature [31, 33]. To create an analogous quantitative metric for causal analysis, we introduce a diagnostic probe. The probe, denoted as $f_{\text{probe}} : \mathbb{R}^{h \times w \times d} \to \mathbb{R}$, is a small convolutional neural network (CNN) trained to classify concepts of interest (e.g., "bird" vs. "cat") based on intermediate model activations.

Specifically, we apply the probe to the output of the conditional forward pass of a chosen cross-attention block, $\boldsymbol{D}^{out}_{t_{\text{probe}}} \in \mathbb{R}^{h \times w \times d}$, at a single, designated diffusion timestep t_{probe}. The metric $\mathcal{M}$ for our circuit discovery is then defined as the probe's raw logit output:

$$\begin{aligned} \mathcal{M}(\boldsymbol{c}) &= \log \mathbb{P}(\text{concept}_{src} | \boldsymbol{D}^{out}_{t_{\text{probe}}}(\boldsymbol{z}_{t_{\text{probe}}}, \boldsymbol{c})) - \log \mathbb{P}(\text{concept}_{dest} | \boldsymbol{D}^{out}_{t_{\text{probe}}}(\boldsymbol{z}_{t_{\text{probe}}}, \boldsymbol{c})) \\ &= f_{\text{probe}}(\boldsymbol{D}^{out}_{t_{\text{probe}}}(\boldsymbol{z}_{t_{\text{probe}}}, \boldsymbol{c})) \end{aligned}$$

This value provides a continuous, differentiable measure of how strongly the model's internal state at t_{probe} represents one concept over the other, enabling the use of the gradient-based attribution methods described above.

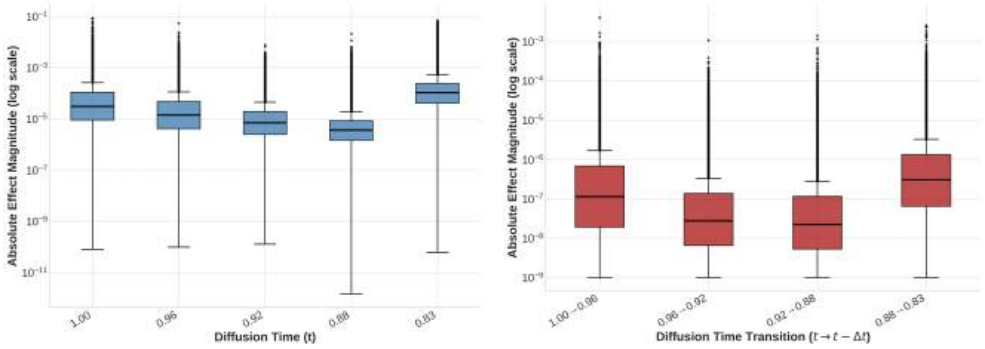
5.3 Anatomy of a 'Cat vs. Bird' Circuit

To ground our theoretical framework in a concrete circuit, we reuse the SAE from Section 4.1 to trace the difference in generation between "cat" and "bird" prompts in the `down.2.0` block. To isolate discriminative features from concept-specific ones, we employ attribution patching in both directions: computing indirect effects when patching bird activations into cat prompts and vice versa. This bidirectional approach helps identify features whose ablation breaks concept generation and whose activation restores it, revealing causal necessity in both directions [33]. We perform this analysis on a curated dataset of prompt pairs, such as "A photorealistic image of a black and white cat facing the camera" versus "A photorealistic image of a black and white bird facing the camera", where only the target concept is altered.

Our analysis focuses on the very early stage ($t \in [0.8, 1.0]$) of the diffusion process, which, as a reminder, is essential for establishing high-level image compositions. The probe $f_{\text{probe}}(\cdot))$ is applied directly to the latent state activations immediately following the cross-attention block at the end of this analysis window. For computational efficiency, the gradient is calculated from the probe's output. This strategically truncates the backward pass, eliminating the need to backpropagate through the computationally expensive remaining steps of the diffusion process. The total indirect effect of each node, which is either an SAE feature or an SAE error component, is computed using Integrated Gradients over 10 steps for improved accuracy. To maintain computational tractability in estimating the edges of the graph, which signify the flow of information, the Jacobian-vector product calculation is restricted to pathways targeting the 10 nodes of the subsequent timestep with the highest absolute indirect effect at each step. The final generalized circuit is constructed by averaging these estimates across 50 prompt pairs, with each pair being processed using two different random seeds.

To render the DAG of all causal interactions tractable, we first identify its most influential components. The quantitative analysis in Figure 5.4 shows that the magnitudes of causal effects vary substantially across diffusion time. Specifically, the most significant indirect effect magnitudes are concentrated in a few timesteps, with peak values in some steps being considerably greater than the peaks in others. This disparity makes any single, global magnitude threshold for pruning the circuit untenable, as it would disproportionately select features from a few high-magnitude timesteps while ignoring the most influential components from other, lower-magnitude stages. Hence, in contrast to prior work that employs a uniform threshold operator to prune circuit graphs [31], we select the top-7 nodes and the top-5 edges independently at each timestep. This approach preserves the integrity of the complete Markov chain by constructing the circuit from the most im-

pactful components at every step, guaranteeing that no computational stage is skipped simply because its local effect magnitudes are smaller than the global peaks.



(a) Node Indirect Effect (IE) Magnitudes. **(b)** Edge Effect Magnitudes.

Figure 5.4: Distribution of Indirect Effect Magnitudes Across Diffusion Time. The boxplots show the distributions of absolute effect magnitudes for **(a)** node indirect effects (IE) and **(b)** edge weights at different stages of the early diffusion process. The y-axis is on a logarithmic scale to accommodate the wide range of values. All indirect effects are aggregated from computations across 50 prompt pairs and two random seeds.

The resulting circuit, illustrated in Figure 5.5, provides a glimpse into the model's internal computations, revealing several key structural patterns in its algorithm. In this circuit, node color signifies a feature's indirect effect on the final classification: red indicates a positive influence (tending to correspond to "bird") and blue a negative influence (tending to correspond to "cat"). A notable characteristic is the temporal persistence of certain features, such as F380, F1761, and F4451, which are active across multiple consecutive timesteps. However, their influence is not static, as their node colors can shift in intensity or even polarity over time. This dynamic behavior may be due to complex indirect effects, where a feature influences downstream components that, in turn, alter the final outcome, or it could result from the aggregation of unconditional and conditional passes, leading to unexpected consequences. Structurally, the circuit also exhibits a scarcity of autoregressive edges (connections from a feature to itself in the next timestep). Instead, the error term (Res) emerges as a critical pathway, whose importance is underscored by its consistent ranking among the top seven nodes at every timestep. The high importance of this residual stream may indicate limitations in the SAE's reconstruction fidelity, suggesting that significant causal information not captured by the sparse features is relegated to and propagated through this channel.

We now analyze the SAE features that demonstrate temporal persistence in the circuit, as their sustained presence suggests a significant role in distin-

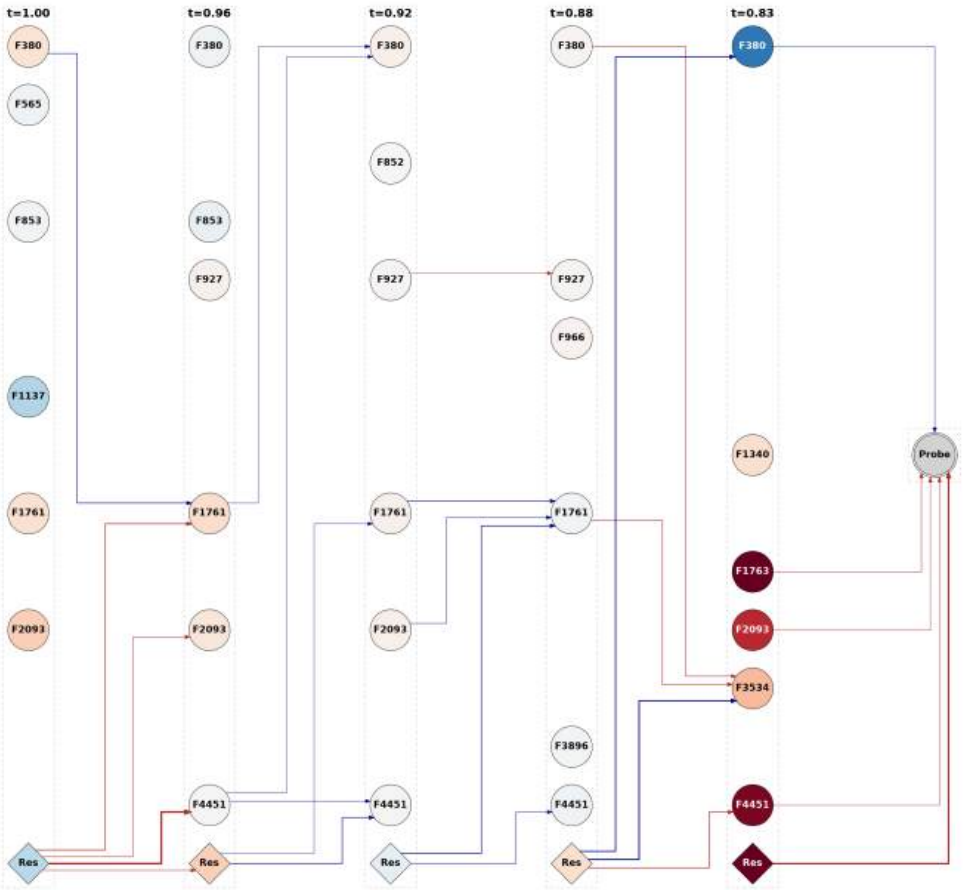


Figure 5.5: Circuit for Distinguishing Cat vs. Bird Generation. The circuit for the `down.2.0` block over time is aggregated across 50 prompts and two random seeds. Nodes are organized into columns by diffusion timestep, for $t \in [0.8, 1.0]$. Circular nodes represent SAE features, while the diamond nodes represent the SAE error terms (Res). Node color indicates the total indirect effect on the final classification (red for positive, blue for negative), with intensity scaling with effect magnitude. Edges represent causal influence, where color indicates the effect's direction (red for positive, blue for negative) and thickness corresponds to its strength. The final "Probe" node is the classifier measuring the outcome.

guishing between cat and bird generation. To understand their function, we visualize their top-activating examples in Figure 5.6 and confirm their causal influence on the final image through single-feature interventions in Figure 5.7. The top-activating images show that the features correspond to distinct concepts: Feature 1761 activates for small birds, 2093 for owls, and 4451 for domestic cats. Feature 853 activates for lions, squirrels, and dogs, while feature 380 activates on dogs, squirrels, and cats. Causal steering experiments confirm these roles, as amplifying these respective features transforms a neutral prompt into corresponding bird- or mammal-like figures. In stark contrast, feature 927 activates on a disparate set of images, such as human faces and glass, and produces abstract, semantically ambiguous results when amplified. Given that Feature 927 lacks a clear connection to either birds or cats in both analyses, we exclude it from further consideration.

Figure 5.6: Top-Activating Images for Circuit Features. Each row displays a circuit feature's top nine most-activating examples from a test set, consisting of 5,000 LAION-COCO prompts across five different seeds, that maximally activate a specific feature of our identified circuit, as measured by the average activation $\bar{\mu}^{\varphi}$. The heatmaps overlaid on the images visualize the sparse activations $S^{\varphi}_{t^{\varphi}_{peak}}$ at the feature's peak activation time, with warmer colors indicating higher values.

To further validate the task-relevance of the discovered circuit nodes, we perform bidirectional activation patching using the method described in Equation 5.2. This intervention swaps the circuit features between concepts by subtracting the destination concept's features while adding those of the source concept for both the "cat" $\rightarrow$ "bird" and "bird" $\rightarrow$ "cat" transfers.

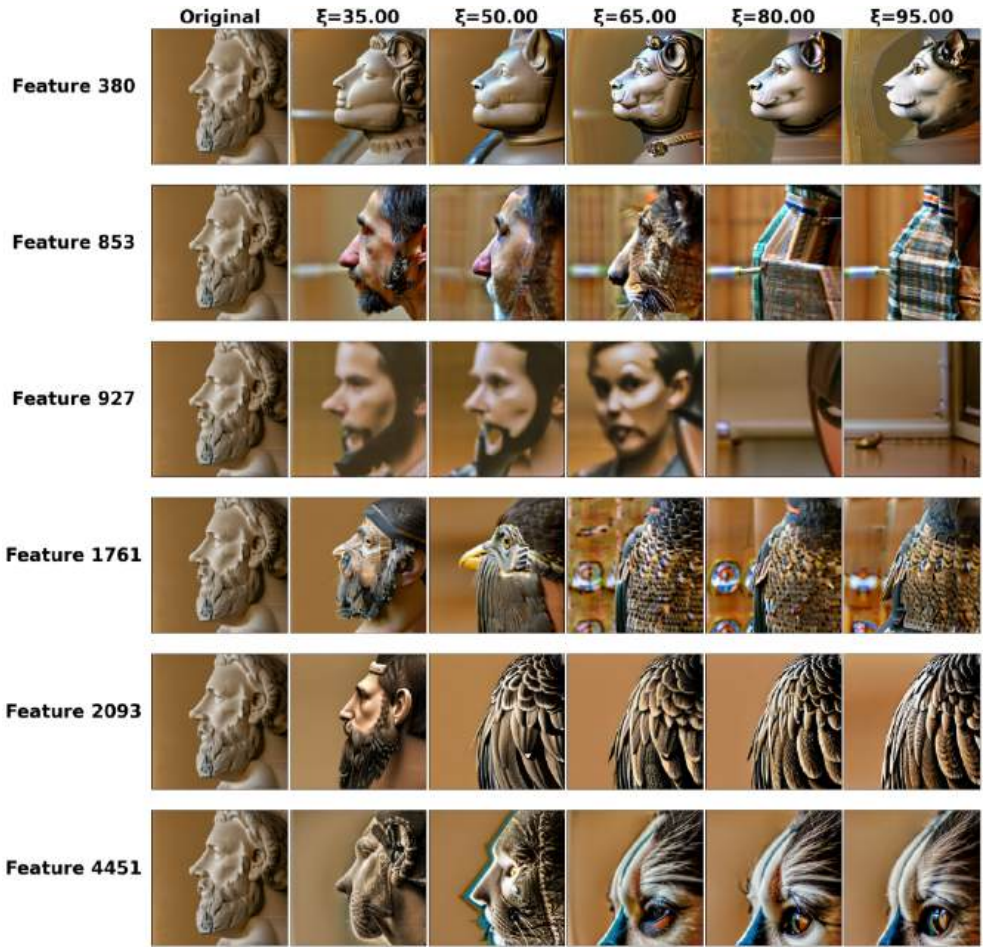


Figure 5.7: Steering Image Generation by Amplifying Circuit Features. All images are generated using the prompt "Portrait of a stoic king, profile view.". Each row demonstrates the effect of amplifying a single feature φ included in the discovered "birds vs cats" circuit at every timestep throughout the generation process. The leftmost column shows the baseline generation without intervention. In the subsequent columns, the feature is amplified by adding $\xi \cdot f_\varphi$ to the cross-attention block's residual update, with strength ξ increasing from 35 to 95.

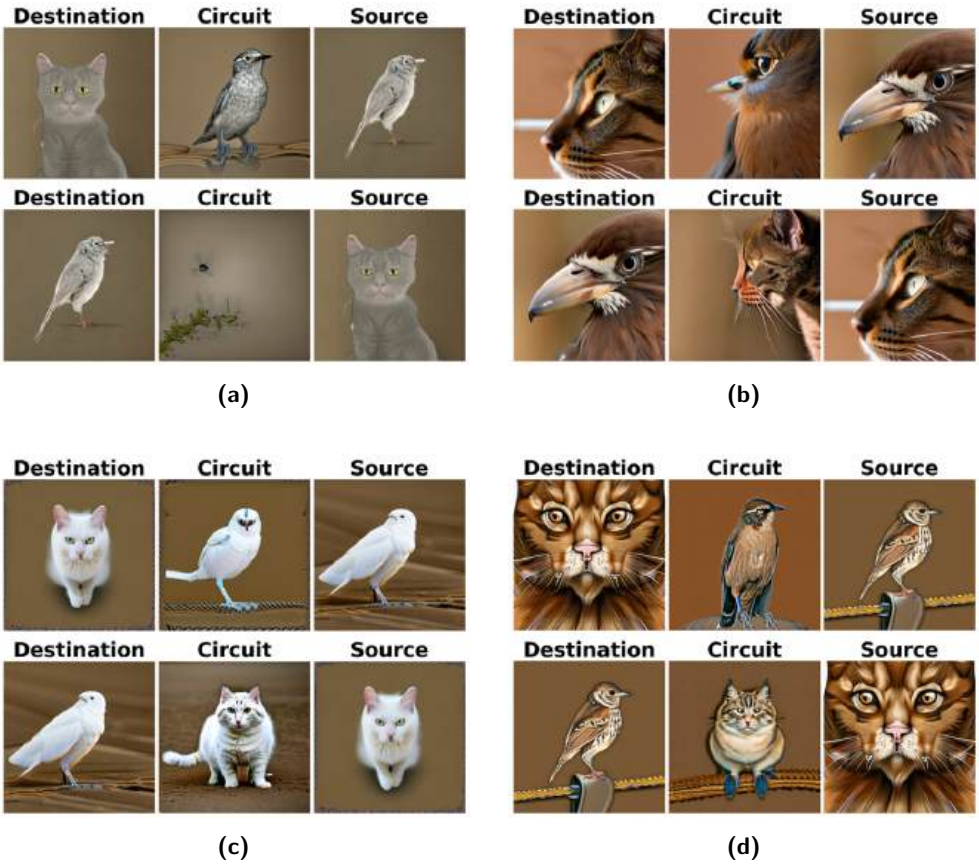


Figure 5.8: Bidirectional Circuit Feature Transfer. Activation patching is performed by adding the identified "cat" features to the "bird" prompt's activations while subtracting the original "bird" features, and vice versa. The subfigures correspond, from top-left to bottom-right, to the following randomly selected prompt templates where [concept] is either "cat" or "bird": (a) "A close-up image of a brown [concept] in profile.", (b) "An artistic image of a grey [concept] sitting still.", (c) "A highly detailed image of a white [concept] facing the camera.", and (d) "A highly detailed image of a brown [concept] sitting still."

Figure 5.8 presents results from four randomly selected test prompt pairs, where this feature exchange reliably transforms the generated subject. The high success rate of these transformations provides strong qualitative evidence that our method isolates the specific neural components that causally control the visual generation of the "bird" versus "cat" concepts, rather than just identifying statistical correlations.

Chapter 6

Discussion

Our work provides a mechanistic analysis of the internal computations in Stable Diffusion v2.1, using a unified sparse autoencoder to deconstruct cross-attention updates into an interpretable feature basis. By intervening at different stages of the generation process, we demonstrate a clear temporal hierarchy in the diffusion process, where the early stages are critical for establishing not only an image's composition but also its core aesthetic. This finding aligns with the established frequency-based understanding [9] of diffusion models, which construct images hierarchically from low-frequency global structures to high-frequency details. Our results both support and extend the work of Tinaz et al. [49]. We concur with their findings that an image's foundational composition is solidified during the early stage ($t \in [0.6, 1.0]$) and that late-stage interventions ($t \in [0.0, 0.2]$) only yield minor textural adjustments. However, our findings on the timeline of style formation diverge. While they identify the mid-stage as the primary window for stylistic control, our qualitative results demonstrate that global style is also established with profound impact during the early stage. This discrepancy may stem from several key methodological differences. Tinaz et al. [49] apply additive feature interventions with equal strength to both conditional and unconditional forward passes, a technique that we and Surkov et al. [46] empirically find can dampen a feature's visual effect compared to the asymmetric approach from Equation 4.1 we employ. Furthermore, their use of separate SAEs for discrete timesteps $t \in \{0.0, 0.5, 1.0\}$ in Stable Diffusion v1.4 contrasts with our unified SAE trained across all timesteps on `down.2.0`, a block known for style and object representation, in the more recent Stable Diffusion v2.1.

Another key finding is the strong cross-resolution generalization of our learned features. An SAE trained exclusively on data from 512×512 image generations maintains its reconstruction fidelity and the causal impact of its features when applied to 768×768 generation. This suggests that

SAEs learn fundamental, potentially almost resolution-invariant, compositional concepts within the diffusion model's representations. This robustness to shifts in the data distribution is further evidenced by recent work [46], which shows that an SAE trained on a single-step distilled model can generalize to multi-step, non-distilled variants. This powerful generalization capability presents a clear path toward more efficient research, allowing SAEs to be trained on computationally cheaper, lower-resolution, or distilled data before being deployed for analysis in more demanding, high-resolution settings.

Moving beyond single-feature analysis, we lay the methodological groundwork for circuit discovery in diffusion models. However, this work has several limitations that frame our contributions and highlight avenues for future research. First, our circuit discovery is demonstrated on a simple comparative task "cat" versus "bird" task within the earliest five diffusion timesteps, and its evaluation remains primarily qualitative. Future work must extend this analysis to more complex and diverse datasets to validate the generalizability of our methods. For instance, applying circuit discovery to datasets designed to probe for hidden social biases would be an exciting and important direction. Crucially, by analyzing only a single cross-attention block, we necessarily capture an incomplete picture of the model's full algorithm. This scope renders a quantitative assessment of the circuit's faithfulness and completeness via mean or zero ablations methodologically unsuitable, as other network components could compensate for the ablated computation. Furthermore, this work currently lacks a direct comparison to neuron-based circuit discovery methods. Additionally, the observation that the SAE's reconstruction error plays a substantial causal role in our analysis might stem from the modest dictionary size and sparsity level ($n_f = 5120$, $k = 10$). We expect a larger SAE with less sparsity would capture more of the model's computations, thereby diminishing this residual effect. However, we note that also in Marks et al. [31], removing SAE error terms is disruptive to the feature circuits. Finally, while we use gradient-based approximations, such as Integrated Gradients [45], to estimate causal effects, a more thorough analysis is needed to validate the quality of these indirect effect estimates. It is imperative to investigate the influence of classifier-free guidance on these estimations for both SAE features and the error term, as our current analysis simply aggregates over both the conditional and unconditional passes.

This work opens several avenues for future research. Building on the promising extrapolation results from our work and Surkov et al. [46], future work should systematically assess the generalizability of SAE features across more diverse scenarios, such as across different model architectures using methods like Sparse Crosscoders [29] or Universal Sparse Autoencoders [48]. For instance, such an analysis could entail comparing the features (or circuits)

learned for a standard multi-step diffusion model to those from distilled or Low-Rank Adaptation (LoRA) [19] fine-tuned variants. Furthermore, mechanistic interpretability tools can be used to contrast the internal computations of legacy models, such as Stable Diffusion v1.4 [37], with newer releases like Stable Diffusion XL [35]. Another promising direction is the development of analysis methods tailored specifically to diffusion models, for example, by explicitly incorporating their temporal or frequency components. Our circuit discovery techniques could also be refined for greater spatial specificity by integrating methods such as spatial masking to isolate analysis to specific image regions. Ultimately, this circuit analysis can be extended across multiple cross-attention blocks to construct a more comprehensive model of how concepts emerge and interact over time throughout the network.

Chapter 7

Conclusion

This thesis presents a mechanistic exploration of Stable Diffusion v2.1, beginning with the decomposition of its internal states into a dictionary of interpretable features. Our feature-level analysis reveals a distinct temporal hierarchy: the generative process is a structured, multi-stage computation where global composition and style are established early, followed by a mid-stage of conceptual refinement and integration, culminating in the later stages, which execute fine-grained textural refinement.

However, the primary contribution of this work lies not merely in identifying these features, but in providing a framework to understand their temporal interplay and elucidate the model's underlying computational mechanisms. Building upon this feature-level foundation, we introduce a novel methodology for temporal circuit discovery in diffusion models. By adapting gradient-based attribution techniques from the language model interpretability literature and validating our findings with a novel bidirectional patching method, we take a foundational step towards mapping the computational graphs that govern the diffusion process. Our initial application of this method to distinguish between "cat" and "bird" concepts uncovers a primitive computational circuit. This circuit-level view moves the analysis from what individual components represent to how they collectively function over time.

This shift in perspective is critical. It demonstrates that complex visual concepts emerge from dynamic, multi-step feature interactions. Understanding these circuits is the key to moving beyond simple, single-feature interventions toward more sophisticated and precise model control. Ultimately, this work contributes to the effort to move beyond a black box view of diffusion models, offering tools and a conceptual framework to analyze them as complex, decomposable systems. Such a detailed mechanistic understanding is essential for the long-term goal of developing reliable, safe, and controllable generative AI.

Appendix A

Background

A.1 Johnson-Lindenstrauss Lemma

Johnson and Lindenstrauss [21] show that any set of n points in a high-dimensional Euclidean space can be projected into a lower-dimensional Euclidean space with only a small distortion of pairwise distances. Concretely, for any given distortion level $\varepsilon \in (0,1)$, one can choose a target dimension $d = \mathcal{O}\left(\frac{\log n}{\varepsilon^2}\right)$ so that all squared pairwise distances are preserved up to a factor of $1 \pm \varepsilon$.

Theorem A.1 (Johnson-Lindenstrauss Lemma) *For any $\varepsilon \in (0,1)$, any integer $n \geq 2$, and any set of points $\mathcal{A} = \{\boldsymbol{x}_i : i \in [n]\} \subset \mathbb{R}^p$, let d be such that*

$$d \geq \frac{4}{\varepsilon^2/2 - \varepsilon^3/3} \log n.$$

Then, there exists a linear map $f : \mathbb{R}^p \to \mathbb{R}^d$ that is an ε–isometry, meaning that

$$(1-\varepsilon)\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 \leq \|f(\boldsymbol{x}_i) - f(\boldsymbol{x}_j)\|^2 \leq (1+\varepsilon)\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2, \quad \forall \boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{A}.$$

This map f can be found in randomized polynomial time.

For a complete proof of the Johnson–Lindenstrauss Lemma, we refer to Dasgupta and Gupta [7].

A.2 Guidance Mechanisms

Conditional generation in diffusion models is achieved through guidance mechanisms that modify the score function to incorporate conditioning information. Dhariwal and Nichol [8] demonstrate that classifier guidance enhances image sample quality by amplifying the conditioning signal. Specifically, they utilize gradients from a classifier that was trained separately on

noisy images to steer the generation. Starting with Bayes' theorem, we have that

$$\nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t|\boldsymbol{c}) = \nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{c}|\boldsymbol{x}_t) + \nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t),$$

where $\boldsymbol{x}_t$ represents the noisy state at timestep t in the diffusion process and $\boldsymbol{c}$ represents conditioning information that guides this generation process. Hence, by introducing an inverse temperature parameter $\omega \geq 0$ that controls the guidance strength, the guided score function becomes

$$\nabla_{\boldsymbol{x}_t} \log p_\omega(\boldsymbol{x}_t|\boldsymbol{c}) = \nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t) + \omega \nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{c}|\boldsymbol{x}_t). \tag{A.1}$$

From the equivalent formulation

$$p_\omega(\boldsymbol{x}_t|\boldsymbol{c}) \propto p(\boldsymbol{x}_t) p(\boldsymbol{c}|\boldsymbol{x}_t)^\omega,$$

it is evident that for larger values of ω, more mass is concentrated on high-likelihood regions of the conditional distribution.

Classifier-free guidance, on the other hand, does not require training a noise-robust classifier model. Instead, Ho and Salimans [18] employ a single model to handle both conditional and unconditional generation. During training, the conditioning information $\boldsymbol{c}$ is randomly dropped out and replaced with a null token $\emptyset$. By applying Bayes' theorem, we can express the conditioning term as

$$\nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{c}|\boldsymbol{x}_t) = \nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t|\boldsymbol{c}) - \nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t).$$

Inserting this into (A.1) yields

$$\nabla_{\boldsymbol{x}_t} \log p_\omega(\boldsymbol{x}_t|\boldsymbol{c}) = \nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t) + \omega(\nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t|\boldsymbol{c}) - \nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t)).$$

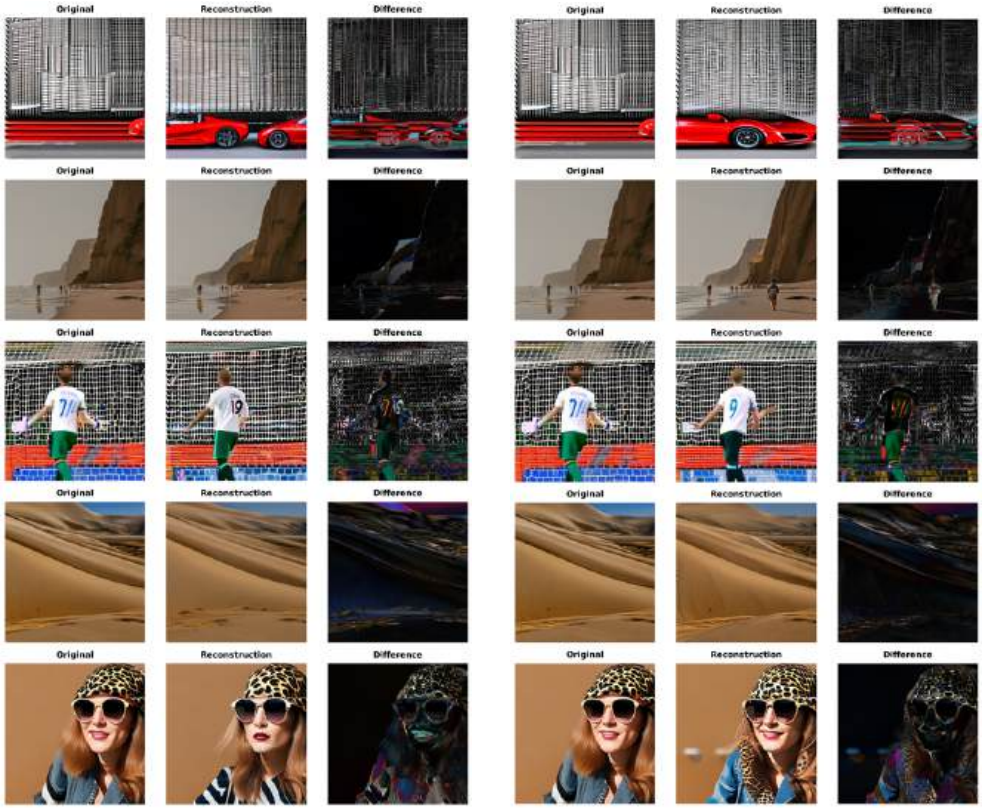
Finally, by parametrizing the score function with a neural network $\boldsymbol{\varepsilon}_\theta(\cdot)$, we obtain classifier-free guided predictions that combine conditional and unconditional estimates:

$$\begin{aligned}\tilde{\boldsymbol{\varepsilon}}_\theta(\boldsymbol{x}_t, t, \boldsymbol{c}) &= \boldsymbol{\varepsilon}_\theta(\boldsymbol{x}_t, t, \emptyset) + \omega \cdot (\boldsymbol{\varepsilon}_\theta(\boldsymbol{x}_t, t, \boldsymbol{c}) - \boldsymbol{\varepsilon}_\theta(\boldsymbol{x}_t, t, \emptyset)) \\ &= \omega \boldsymbol{\varepsilon}_\theta(\boldsymbol{x}_t, t, \boldsymbol{c}) + (1-\omega)\boldsymbol{\varepsilon}_\theta(\boldsymbol{x}_t, t, \emptyset),\end{aligned}$$

where for $\omega > 1$, the generated samples more strongly adhere to the conditioning. Finally, at test time, for any positive guidance scale $\omega \notin \{0, 1\}$, this formulation necessitates two forward passes of the network per sampling step: one for the conditional estimate $\boldsymbol{\varepsilon}_\theta(\boldsymbol{x}_t, t, \boldsymbol{c})$ and another for the unconditional estimate $\boldsymbol{\varepsilon}_\theta(\boldsymbol{x}_t, t, \emptyset)$.

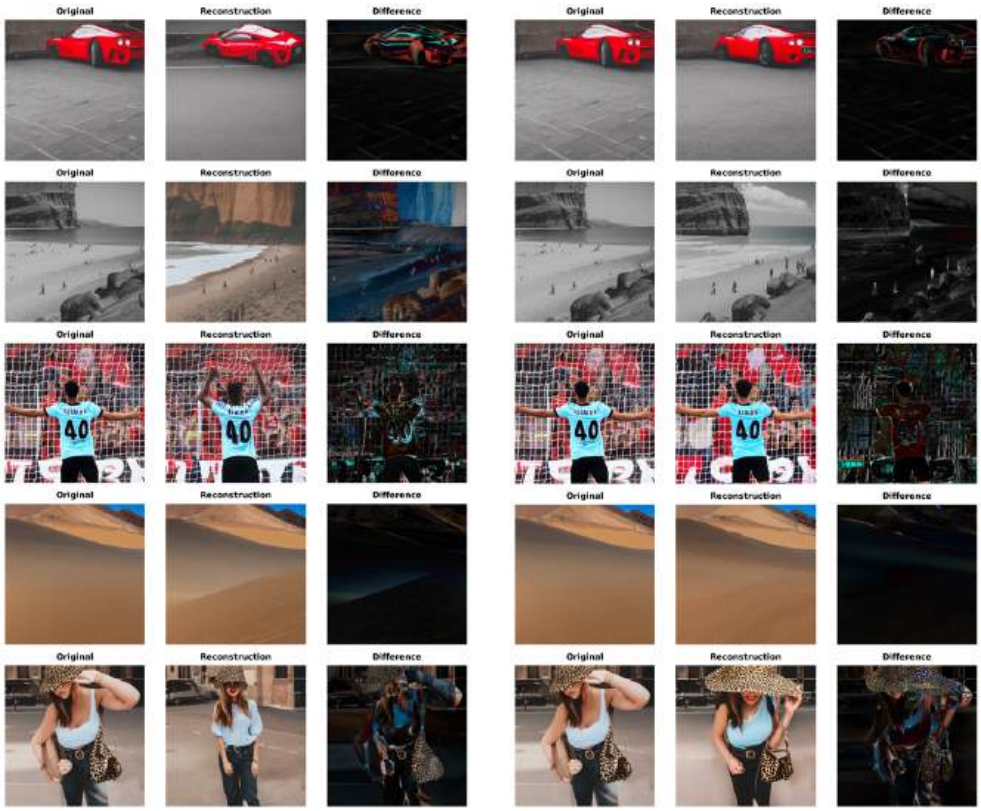
Appendix B

Complementary Visualizations



(a) `down.2.0` **(b)** `up.1.1`

Figure B.1: Sparse Autoencoder Reconstructions. Comparison of reconstructions for five random images from the LAION-COCO dataset. The panels display reconstruction outputs from two distinct sparse autoencoders, each trained independently on the activations of a different cross-attention block within the denoising U-Net.



(a) `down.2.0` **(b)** `up.1.1`

Figure B.2: High-Resolution Sparse Autoencoder Reconstructions. Comparison of reconstructions leveraging an SAE (trained on latents from 512×512 generations) for five random images (resolution 768×768) from the LAION-COCO dataset. The panels display reconstruction outputs from two distinct sparse autoencoders, each trained independently on the activations of a different cross-attention block within the denoising U-Net.

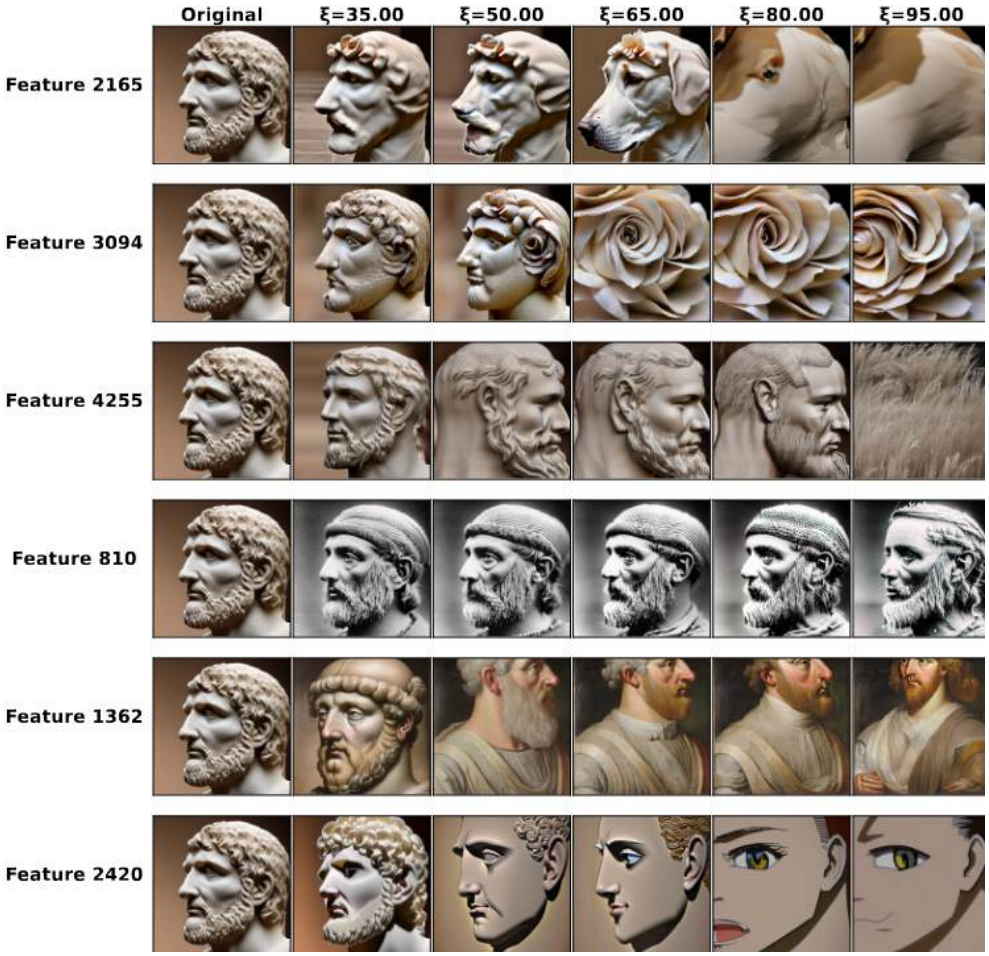


Figure B.3: Steering High-Resolution Image Generation by Amplifying Individual Features. All images (resolution 768×768) are generated using the prompt "Portrait of a stoic Roman emperor, profile view.". Each row demonstrates the effect of amplifying a single feature φ at every timestep throughout the generation process. The leftmost column shows the baseline generation without intervention. In the subsequent columns, the feature is amplified by adding $\xi \cdot \boldsymbol{f}_\varphi$ to the `down.2.0` block's residual update, with strength ξ increasing from 35 to 95.

Figure B.4: Comparison of Classifier-Free Guidance Interventions. Activation patching results transferring features from the source prompt "A high-resolution image of a rose." to the destination prompt "A high-resolution image of a dog.". The transfer is achieved by subtracting the top-m features most characteristic of the control concept (with a scaling factor of $\alpha = 1$) and adding the top-m features most characteristic of the source concept (with a scaling factor of $\beta = 2$). Columns show the effect of patching an increasing number of features ($m \in \{1, 5, 20, 50, 200\}$). The rows, from top to bottom, show: an additive intervention on the unconditional pass only **(+uncond)**; on the conditional pass only **(+cond)**; on both passes **(+uncond, +cond)**; and a subtractive intervention on the unconditional pass while adding to the conditional pass **(-uncond, +cond)**.

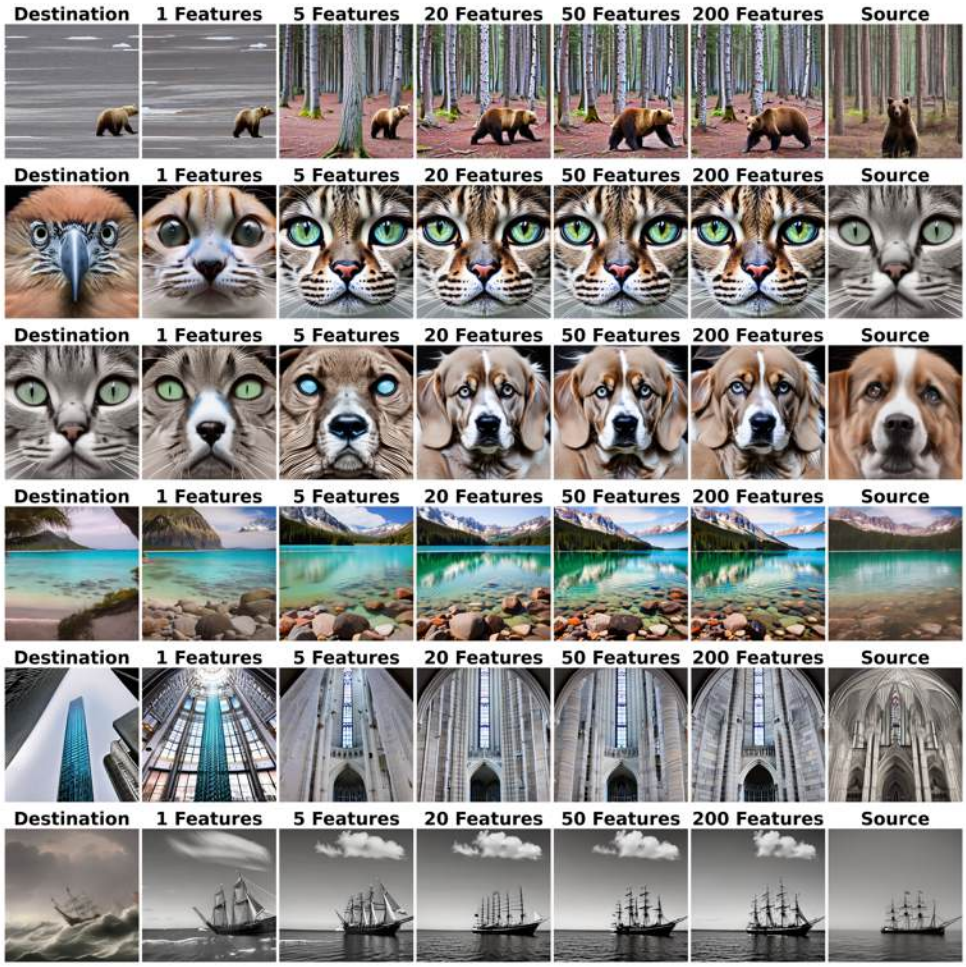


Figure B.5: High-Resolution Feature Activation Transfer in down.2.0. Activation patching results when using an image resolution of 768×768, while leveraging an SAE trained on latents from 512×512 generations. The transfer is achieved by subtracting the top-m features most characteristic of the control concept (with a scaling factor of $\alpha = 1$) and adding the top-m features most characteristic of the source concept (with a scaling factor of $\beta = 2$). Columns show the effect of transplanting an increasing number of feature activations ($m \in \{1, 5, 20, 50, 200\}$) from a source to a control concept. From top to bottom, the rows demonstrate the following conceptual transfers: "forest" in place of "arctic"; "cat" in place of "bird"; "dog" in place of "bird"; "mountain lake" in place of "ocean bay"; "cathedral" in place of "glass skyscraper"; "calm ocean" in place of "rough ocean".

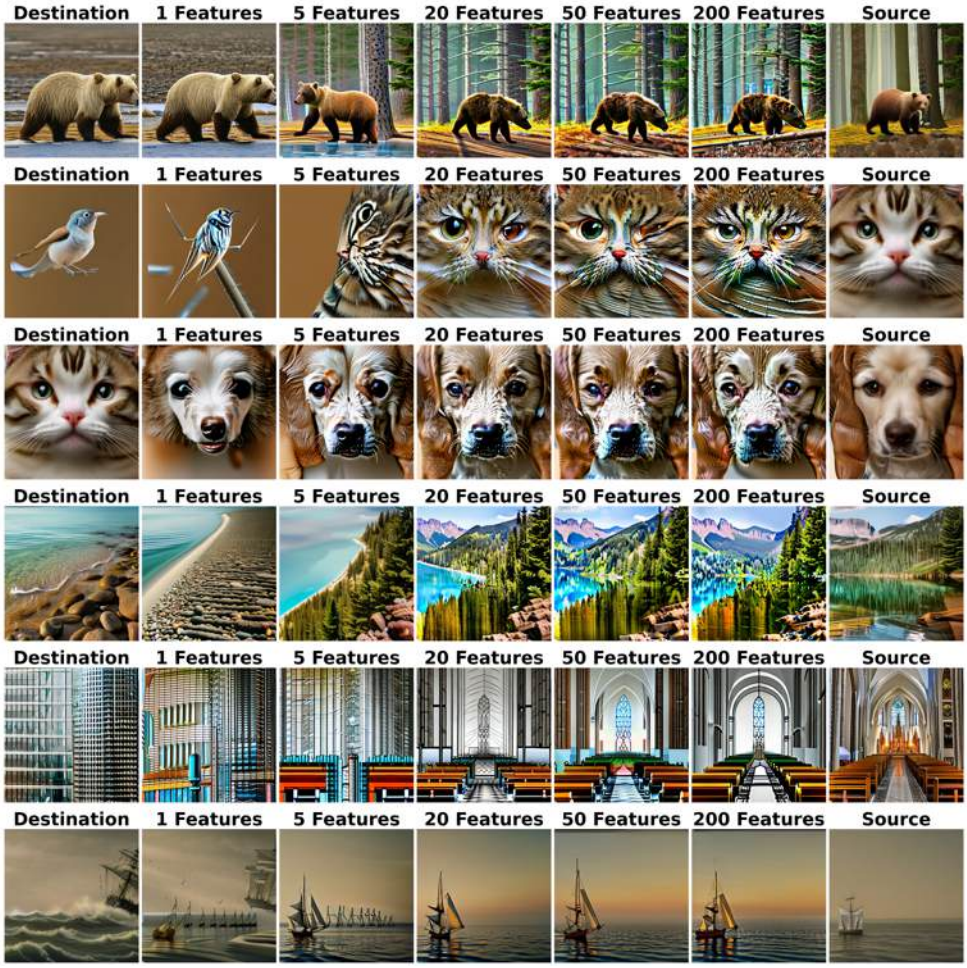


Figure B.6: Feature Activation Transfer in up.1.1. Activation patching is performed in the `up.1.1` cross-attention block across all diffusion timesteps $t \in [0, 1]$. The transfer is achieved by subtracting the top-m features most characteristic of the control concept (with a scaling factor of $\alpha = 1$) and adding the top-m features most characteristic of the source concept (with a scaling factor of $\beta = 2$). Columns show the effect of transplanting an increasing number of feature activations ($m \in \{1, 5, 20, 50, 200\}$) from a source to a control concept. From top to bottom, the rows demonstrate the following conceptual transfers: "forest" in place of "arctic"; "cat" in place of "bird"; "dog" in place of "bird"; "mountain lake" in place of "ocean bay"; "cathedral" in place of "glass skyscraper"; "calm ocean" in place of "rough ocean".

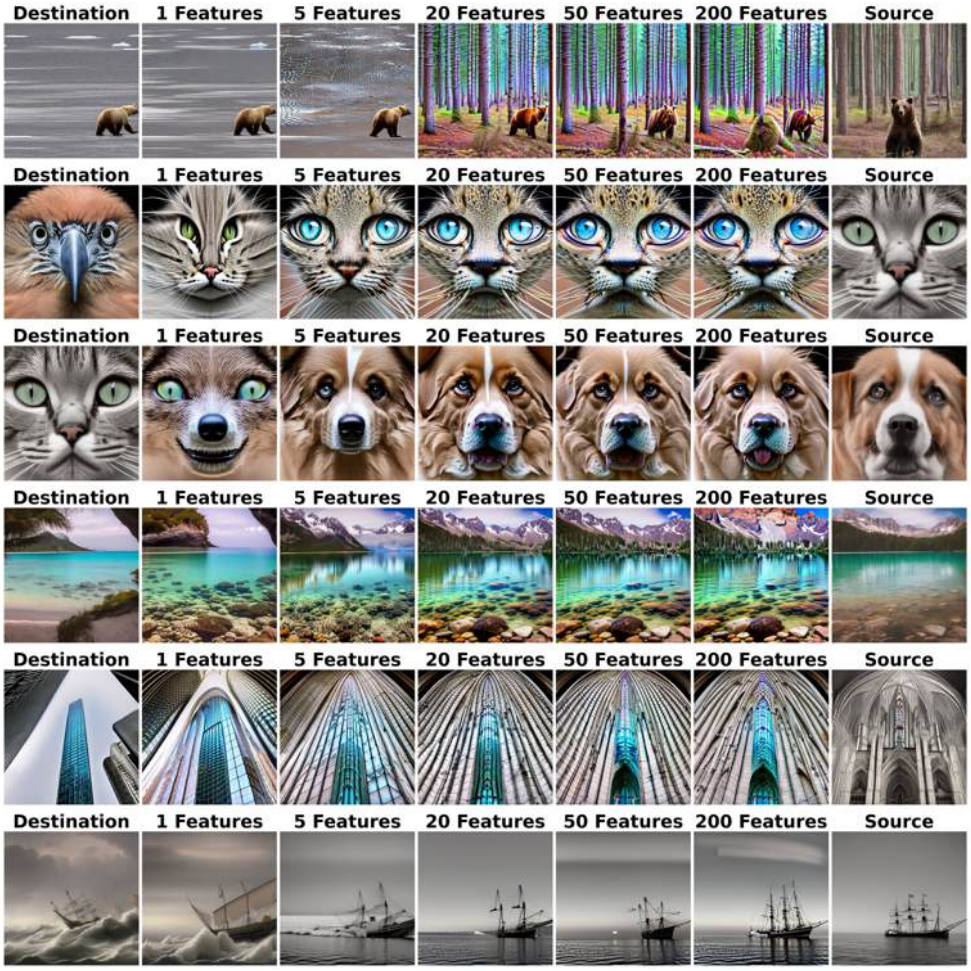


Figure B.7: High-Resolution Feature Activation Transfer in up.1.1. Activation patching results when using an image resolution of 768×768, while leveraging an SAE trained on latents from 512×512 generations.The transfer is achieved by subtracting the top-m features most characteristic of the control concept (with a scaling factor of $\alpha = 1$) and adding the top-m features most characteristic of the source concept (with a scaling factor of $\beta = 2$). Columns show the effect of transplanting an increasing number of feature activations ($m \in \{1, 5, 20, 50, 200\}$) from a source to a control concept. From top to bottom, the rows demonstrate the following conceptual transfers: "forest" in place of "arctic"; "cat" in place of "bird"; "dog" in place of "bird"; "mountain lake" in place of "ocean bay"; "cathedral" in place of "glass skyscraper"; "calm ocean" in place of "rough ocean".

Bibliography

[1] Emmanuel Ameisen, Jack Lindsey, Adam Pearce, Wes Gurnee, Nicholas L. Turner, Brian Chen, Craig Citro, David Abrahams, Shan Carter, Basil Hosmer, Jonathan Marcus, Michael Sklar, Adly Templeton, Trenton Bricken, Callum McDougall, Hoagy Cunningham, Thomas Henighan, Adam Jermyn, Andy Jones, Andrew Persic, Zhenyi Qi, T. Ben Thompson, Sam Zimmerman, Kelley Rivoire, Thomas Conerly, Chris Olah, and Joshua Batson. Circuit tracing: Revealing computational graphs in language models. *Transformer Circuits Thread*, 2025.

[2] Samyadeep Basu, Keivan Rezaei, Priyatham Kattakinda, Ryan Rossi, Cherry Zhao, Vlad Morariu, Varun Manjunatha, and Soheil Feizi. On mechanistic knowledge localization in text-to-image generative models, 2024.

[3] Leonard Bereska and Efstratios Gavves. Mechanistic interpretability for ai safety – a review, 2024.

[4] Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. https://transformer-circuits.pub/2023/monosemantic-features/index.html.

[5] Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models, 2023.

[6] Bartosz Cywiński and Kamil Deja. Saeuron: Interpretable concept unlearning in diffusion models with sparse autoencoders, 2025.

[7] Sanjoy Dasgupta and Anupam Gupta. An elementary proof of a theorem of johnson and lindenstrauss. *Random Structures & Algorithms*, 22(1):60–65, 2003.

[8] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis, 2021.

[9] Sander Dieleman. Diffusion is spectral autoregression, 2024.

[10] J. Dunefsky, P. Chlenski, and N. Nanda. Transcoders enable fine-grained interpretable circuit analysis for language models. lesswrong, 2024. Accessed: 2025-09-02.

[11] Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. Toy models of superposition, 2022.

[12] Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021. https://transformer-circuits.pub/2021/framework/index.html.

[13] Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders, 2024.

[14] Liv Gorton. The missing curve detectors of inceptionv1: Applying sparse autoencoders to inceptionv1 early vision, 2024.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[16] Stefan Heimersheim and Neel Nanda. How to use and interpret activation patching, 2024.

[17] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020.

[18] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022.

[19] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021.

[20] Victor Shea-Jay Huang, Le Zhuo, Yi Xin, Zhaokai Wang, Peng Gao, and Hongsheng Li. Tide : Temporal-aware sparse autoencoders for interpretable diffusion transformers in image generation, 2025.

[21] William B Johnson, Joram Lindenstrauss, et al. Extensions of lipschitz mappings into a hilbert space. *Contemporary mathematics*, 26(189-206):1, 1984.

[22] Dahye Kim and Deepti Ghadiyaram. Concept steerers: Leveraging k-sparse autoencoders for controllable generations, 2025.

[23] Dahye Kim, Xavier Thomas, and Deepti Ghadiyaram. *Revelio*: Interpreting and leveraging semantic information in diffusion models, 2024.

[24] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[25] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022.

[26] Xianghao Kong, Ollie Liu, Han Li, Dani Yogatama, and Greg Ver Steeg. Interpretable diffusion via information decomposition, 2024.

[27] János Kramár, Tom Lieberum, Rohin Shah, and Neel Nanda. Atp*: An efficient and scalable method for localizing llm behaviour to components, 2024.

[28] Black Forest Labs. Flux. https://github.com/black-forest-labs/flux, 2024.

[29] Jack Lindsey, Adly Templeton, Jonathan Marcus, Thomas Conerly, Joshua Batson, and Christopher Olah. Sparse crosscoders for cross-layer features and model diffing, 2024.

[30] Alireza Makhzani and Brendan Frey. k-sparse autoencoders, 2014.

[31] Samuel Marks, Can Rager, Eric J. Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models, 2025.

[32] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt, 2023.

[33] Neel Nanda. Attribution patching: Activation patching at industrial scale. https://www.neelnanda.io/mechanistic-interpretability/attribution-patching, 2022. Accessed: 2025-09-03.

[34] Judea Pearl. Direct and indirect effects. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, UAI'01, page 411–420, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[35] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis, 2023.

[36] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021.

[37] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022.

[38] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.

[39] Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation, 2023.

[40] Adam Scherlis, Kshitij Sachan, Adam S. Jermyn, Joe Benton, and Buck Shlegeris. Polysemanticity and capacity in neural networks, 2025.

[41] Christoph Schuhmann, Andreas Köpf, Richard Vencu, Theo Coombes, Romain Beaumont, and Benjamin Trom. Laion coco: 600m synthetic captions from laion2b-en. LAION Blog, 2022. Accessed: 2025-08-12.

[42] Yingdong Shi, Changming Li, Yifan Wang, Yongxiang Zhao, Anqi Pang, Sibei Yang, Jingyi Yu, and Kan Ren. Dissecting and mitigating diffusion bias via mechanistic interpretability, 2025.

[43] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models, 2022.

[44] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations, 2021.

[45] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks, 2017.

[46] Viacheslav Surkov, Chris Wendler, Antonio Mari, Mikhail Terekhov, Justin Deschenaux, Robert West, Caglar Gulcehre, and David Bau. One-step is enough: Sparse autoencoders for text-to-image diffusion models, 2025.

[47] Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, and Tom Henighan. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Transformer Circuits Thread*, 2024.

[48] Harrish Thasarathan, Julian Forsyth, Thomas Fel, Matthew Kowal, and Konstantinos Derpanis. Universal sparse autoencoders: Interpretable cross-model concept alignment, 2025.

[49] Berk Tinaz, Zalan Fabian, and Mahdi Soltanolkotabi. Emergence and evolution of interpretable concepts in diffusion models, 2025.

[50] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.

[51] Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in gpt-2 small, 2022.

[52] Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion models: A comprehensive survey of methods and applications, 2024.

[53] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric, 2018.

[54] Yihua Zhang, Chongyu Fan, Yimeng Zhang, Yuguang Yao, Jinghan Jia, Jiancheng Liu, Gaoyuan Zhang, Gaowen Liu, Ramana Rao Kompella, Xiaoming Liu, and Sijia Liu. Unlearncanvas: Stylized image dataset for enhanced machine unlearning evaluation in diffusion models, 2024.

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every written paper or thesis authored during the course of studies. **In consultation with the supervisor**, one of the following two options must be selected:

☐ I hereby declare that I authored the work in question independently, i.e. that no one helped me to author it. Suggestions from the supervisor regarding language and content are excepted. I used no generative artificial intelligence technologies[1].

☒ I hereby declare that I authored the work in question independently. In doing so I only used the authorised aids, which included suggestions from the supervisor regarding language and content and generative artificial intelligence technologies. The use of the latter and the respective source declarations proceeded in consultation with the supervisor.

Title of paper or thesis:

Mechanistic Interpretability in Text-to-Image Diffusion Models

Authored by:
If the work was compiled in a group, the names of all authors are required.

Last name(s):	**First name(s):**
Schlegel	Jan Heinrich

With my signature I confirm the following:
- I have adhered to the rules set out in the Citation Guidelines.
- I have documented all methods, data and processes truthfully and fully.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for originality.

Place, date	**Signature(s)**
Zurich, 08.09.2025	

If the work was compiled in a group, the names of all authors are required. Through their signatures they vouch jointly for the entire content of the written work.

[1] For further information please consult the ETH Zurich websites, e.g. https://ethz.ch/en/the-eth-zurich/education/ai-in-education.html and https://library.ethz.ch/en/researching-and-publishing/scientific-writing-at-eth-zurich.html (subject to change).